МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ
МІСЬКОГО ГОСПОДАРСТВА

М. І. Самойленко, А. О. Кобець

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В РОЗВ'ЯЗАННІ ТРАНСПОРТНИХ ЗАДАЧ

МОНОГРАФІЯ

За редакцією М. І. Самойленка

ХАРКІВ

ХНАМГ

2011

УДК 656:519.85
ББК 39.18+22.18
С17

Самойленко Микола Іванович
Кобець Анна Олександрівна

Рецензенти:
Колосов Анатолій Іванович**,** завідувач кафедри вищої математики Харківської національної академії міського господарства, доктор фізико-математичних наук, професор.
Левикін Віктор Макарович**,** завідувач кафедри інформаційних управлінських систем Харківського національного університету радіоелектроніки, доктор технічних наук, професор.

Рекомендована до друку Вченою Радою ХНАМГ,
протокол №5 від 28.01.2011 р.

Самойленко М.І.
С17 Інформаційні технології в розв'язанні транспортних задач: монографія. / М. І. Самойленко, А. О. Кобець; Харк. нац. акад. міськ. госп-ва. – Х.: ХНАМГ, 2011. – 256 с.
ISBN 978–966–695–204–5

Наводяться математичні моделі та методи вирішення задач організації перевезень дрібного вантажу. Особлива увага приділяється задачі комівояжера великої вимірності та задачі розбиття транспортної мережі на задану кількість районів.

Для наукових і інженерно-технічних працівників науково-дослідницьких, проектних та виробничих установ, що пов'язані з проектуванням та організацією транспортних перевезень.

УДК 656:519.85
ББК 39.18+22.18

ISBN 978–966–695–204–5

Зміст

Передмова 6

Вступ 7

РОЗДІЛ 1
МАТЕМАТИЧНІ МОДЕЛІ В ЗАДАЧАХ ОРГАНІЗАЦІЇ І ПЛАНУВАННЯ ВАНТАЖНИХ ПЕРЕВЕЗЕНЬ 18

1.1. Задачі зі схемою перевезень типу «один-до-одного». 22
1.1.1. Задача про вибір транспортних засобів і розподіл вантажу. 22
1.1.2. Задача про формування парку машин та їх розподіл. 29
1.2. Задачі зі схемою перевезень «один-до-багатьох». . . 36
1.2.1. Задача про розподіл транспортних засобів за витратами часу. 37
1.2.2. Задача про розподіл транспортних засобів за обсягом перевезень. 43
1.2.3. Задача про розподіл транспортних засобів за грошовими витратами 47
1.2.4. Задача про розподіл транспортних засобів із фіксованими доплатами. 52
1.2.5. Задача про формування парку машин та їх розподіл . 60
1.2.6. Задача про розвезення вантажу. 66
1.2.7. Задача про вибір маршруту. 71
1.3. Задачі зі схемою перевезень типу «багато-до-одного». 78
1.3.1. Задача про планування випуску продукції. 80
1.3.2. Змістовні постановки задач зі схемою перевезень типу «багато-до-одного» 85
1.4. Задачі зі схемою перевезень типу «багато-до-багатьох». 90
1.4.1. Класична транспортна задача 91

1.4.2. Транспортна задача із неперервною відкритою математичною моделлю 98
1.4.3. Транспортна задача із цілочисловою математичною моделлю 102
1.4.4. Транспортна задача про перевезення вантажу за два етапи 108
1.4.5. Транспортна задача про перевезення вантажу декількох видів за два етапи 117
1.4.6. Транспортна задача про перевезення вантажу декількох видів на запити споживачів за два етапи. . . . 119
1.4.7. Транспортна задача про закриття заводу 122
1.4.8. Транспортна задача про розиграш кубка 129

РОЗДІЛ 2
РОЗБИТТЯ ТРАНСПОРТНОЇ МЕРЕЖІ НА РАЙОНИ . 137
2.1. Аналіз підходів до розв'язання задачі розбиття транспортної мережі на райони. 137
2.2. Методика проведення дослідження 144
2.3. Адаптація методу «гілок і границь» до розв'язання задачі розбиття транспортної мережі на райони . . . 146
2.3.1. Алгоритм модифікованого методу «гілок і границь» 156
2.3.2. Перший варіант алгоритму модифікованого методу «гілок і границь» 164
2.3.3. Другий варіант алгоритму модифікованого методу «гілок і границь» 165
2.3.4. Третій варіант алгоритму модифікованого методу «гілок і границь». 167
2.3.5. Четвертий варіант модифікованого методу «гілок і границь» 169
2.3.6. П'ятий варіант алгоритму модифікованого методу «гілок і границь» 173
2.3.7. Шостий варіант алгоритму модифікованого методу «гілок і границь» 174
2.4. Аналіз варіантів алгоритму модифікованого методу «гілок і границь» 178

РОЗДІЛ 3
РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА ВЕЛИКОЇ ВИМІРНОСТІ ЗА ПРИНЦИПАМИ ДИНАМІЧНОГО ПРОГРАМУВАННЯ **181**

3.1. Аналіз існуючих методів маршрутизації для перевезень дрібного вантажу. 182
3.2. Метод динамічного програмування та аналіз його ефективності 196
3.2.1. Формування матриці відстаней між пунктами транспортної мережі. 196
3.2.2. Алгоритм методу динамічного програмування стосовно розв'язання задачі комівояжера 199
3.2.3 Тестування методу динамічного програмування стосовно розв'язання задачі комівояжера. 206
3.3. Модифікований метод динамічного програмування та оцінка його ефективності. 209
3.4. Тестування модифікованого методу динамічного програмування. 215

РОЗДІЛ 4
ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ ДЛЯ ВИЗНАЧЕННЯ МАРШРУТУ КОМІВОЯЖЕРА ТА РОЗБИТТЯ ТРАНСПОРТНОЇ МЕРЕЖІ НА РАЙОНИ. **221**

4.1. Програмна реалізація модифікованого методу динамічного програмування для розв'язання задачі комівояжера 222
4.2. Опис програми, що реалізує алгоритм розбиття транспортної мережі на райони 231
4.3. Приклад застосування програми розбиття транспортної мережі на райони. 242
4.4. Висновки за результатами тестування програм, що реалізують модифікації методів «гілок і границь» і динамічного програмування 248

Список джерел 252

Присвячується 75-річчю з дня народження
Євдокимова Анатолія Гавриловича,
Заслуженого діяча науки і техніки України,
Лауреата державної премії України

Передмова

Предметом монографії є транспортні задачі організації перевезень дрібного вантажу, тобто будуть розглядатися транспортні задачі великої вимірності.

Основна увага в монографії приділяється математичним і цифровим моделям оптимізаційних задач, що зустрічаються в інженерній практиці пасажирських і вантажних перевезень. Для кожного типу задач організації перевезень наводяться загальні змістовні і математичні постановки, конкретні приклади задач та їх розв'язання.

Автори свідомо вибирали для більшості задач приклади з малою вимірністю з метою підсилити наочність і спростити розгляд їх математичного та цифрового моделювання. Виключення складають тільки задача комівояжера та задача розбиття транспортної мережі на задане число районів. Кожна з цих задач займає цілий розділ монографії та орієнтована на велику вимірність мережі. Тут на перший план виходять не стільки самі моделі задач, скільки методи їх розв'язання.

Монографія складається з вступу та чотирьох розділів.

Вступ – це результат спільної роботи авторів. Автором розділу 1 є М. І. Самойленко. Розділи 2, 3 і 4 написані А. О. Кобець і відображують основні результатами її дисертаційних досліджень.

Автори

Вступ

В останні роки в багатьох галузях народного господарства для раціональної організації виробництва і виконання виробничих завдань широко застосовується системний підхід. При такому підході організація окремих етапів виробництва повинна розглядатися з точки зори їх впливу на ефективність усього виробництва в цілому. Організація автомобільних перевезень вантажів не є виключенням: комплексний характер сучасних досліджень з організації перевезень вантажів істотно відрізняє їх від концепції досліджень минулого століття.

З точки зору логістики, транспортні перевезення вантажів являють собою складний процес, що містить в собі етапи планування, організації і виконання доставки вантажу, підготовки партій відправлень до перевезення, організації і проведення вантажно-розвантажувальних робіт, розфасовки, упаковки і складування вантажу, страхування перевезень, іноді – митні послуги. Важливими складовими організації транспортних перевезень є також раціональний вибір транспортних засобів, найбільш повне використання вантажопідйомності транспортних засобів, дотримання технологій вантажно-розвантажувальних робіт, розрахунок необхідних запасів товару для забезпечення безперебійного навантаження та ін. У випадку, коли підприємство-виробник користується послугами перевізника, до елементів транспортного процесу додаються оформлення необхідних перевізних документів, складання договору на перевезення, розрахунок за послуги перевізника. Практична реалізація всіх перерахованих складових транспортного процесу вимагає відповідних грошових і часових витрат. Природно, ефективність перевезень і якість доставки товару будуть залежати від кожної складової транспортного процесу [1, 2, 3]. Тільки оптимальний розподіл матеріальних, трудових і фінансових ресурсів між усіма складовими може гарантувати найвищу ефективність транспортного процесу в цілому.

Оптимальна організація окремих етапів не є умовою оптимальності всього процесу. Однак уміння визначати оптимально можливі результати на будь-якому етапі дозволяє забезпечити дослідників і виробничників важливим критерієм для оцінки впливу цього етапу на ефективність усього процесу. Тому проблема оптимізації кожного етапу транспортного процесу завжди буде залишатися актуальною, але не завжди – визначальною.

Основними учасниками процесу перевезень є виробники, експедитори, торгові фірми і перевізники. Слід враховувати, що в процесі перевезень кожний з учасників переслідує свої цілі, має власні прибутки та витрати.

Розглянемо прибутки і витрати всіх учасників перевезень у рамках процесу транспортування товару, абстрагуючись від його виробництва та продажу.

Прибутками експедитора є оплата фірмою-виробником доставки товару та штрафи. Витрати експедитора – це витрати на пошук перевізника, оплата вартості перевезення, витрати на доставку, штрафи і т.п.

Прибутками перевізника є оплата тарифу перевезення, компенсації за змушену затримку або простою рухомого складу. Витрати складаються із знижок експедитору, витрат на підготовку та здійснення процесу перевезення.

Прибутками торгової фірми є штрафи, оговорені в контракті.

Модель учасників системи доставки може бути подана у вигляді кортежу

$$SD = \langle P, EXP, PER, TF \rangle , \qquad (0.1)$$

де $P = P(x) = \{x_1, x_2, ..., x_n\}$ – множина виробників, $EXP = EXP(y) = \{y_1, y_2, ..., y_m\}$ – множина експедиторів, $PER = PER(z) = \{z_1, z_2, ..., z_k\}$ – множина перевізників; $TF = TF(t) = \{t_1, t_2, ..., t_l\}$ – множина торгових фірм. Тут має місце процес доставки, у якому беруть участь n виробників, m експедиторів, k перевізників і l торгових фірм.

У загальному випадку цільова функція системи доставки, що визначає якість товару та витрати коштів і часу на його перевезення, має вигляд

$$E(SD) \to \min . \tag{0.2}$$

Якість системи визначається надійністю доставки, своєчасністю і тривалістю транспортування вантажів, транспортними витратами, гнучкістю перебудови схеми транспортування відповідно до потреб споживачів, схоронністю вантажу, швидкістю підготовки документів, витратами на кріплення, завантаження і складування вантажів, витратами на страхування перевезення.

Узагальнена модель (0.1) може бути використана для дослідження якості системи доставки за умови її конкретизації більш точними математичними виразами, що описують усі умови процесу, прибутки та витрати всіх учасників перевезень: і виробників, і експедиторів, і перевізників, і споживачів. У роботі [4] наведений алгоритм планування вантажних автомобільних перевезень, що відповідає такій моделі. Він полягає у виконанні наступних етапів:

1. *Формування бази даних*. База даних повинна містити відомості про:

– кількість вантажовідправників і вантажоодержувачів;

– об'єми вантажу на складах вантажовідправників і його основні характеристики;

– потреби кожного вантажоодержувача в об'ємах вантажу;

– кількість, тип і вантажопідйомність транспортних засобів;

– обмеження, що накладаються вантажовідправником і вантажоодержувачем на партію вантажу, що може бути відправлений або отриманий відповідним суб'єктом;

– обмеження в часі на доставку вантажів у пункти призначення або вивозу вантажу з пунктів відправлення;

– відстані між пунктами навантаження і розвантаження;

– питомі тимчасові витрати на переміщення вантажу на одиницю відстані на кожній ділянці маршруту;

– питомі матеріальні витрати на переміщення одиниці вантажу на одиницю відстані;

– витрати часу на навантаження і розвантаження;

– можливі затримки в часі у процесі проходження маршруту та інші.

2. *Визначення типу схеми організації перевезень*. Визначення робиться на підставі отриманих замовлень на доставку. Схеми перевезень підрозділяються за типами, а саме: «один-до-одного», «один-до-багатьох», «багато-до-одного» та «багато-до-багатьох». Схема типу «один-до-одного» припускає, що від одного постачальника (з одного пункту відправлення) необхідно доставити вантаж тільки одному замовнику (в один пункт призначення). Під схемою типу «один-до-багатьох» мається на увазі, що доставка буде виконуватися від одного постачальника та доставлятися багатьом замовникам. Схема «багато-до-одного» припускає перевезення від багатьох постачальників тільки до одного замовника. При організації перевезень за схемою «багато-до-багатьох» вантаж постачається від декількох виробників або декількох складів одного виробника декільком замовникам [5].

3. *Моделювання процесу перевезення*. У найбільш загальному випадку перевезення вантажу здійснюється за схемою «багато-до-багатьох», тому в першу чергу увага дослідників приділя-

ється саме математичній моделі перевезень цього типу. Формальна постановка класичної транспортної задачі має вигляд:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}\cdot x_{ij} \to \min_{x_{ij}\in\Omega}, \tag{0.3}$$

$$\Omega:\quad \sum_{j=1}^{m} x_{ij} = a_i, \quad i=\overline{1,n}, \tag{0.4}$$

$$\sum_{i=1}^{n} x_{ij} = b_j, \quad j=\overline{1,m}, \tag{0.5}$$

$$x_{ij} \ge 0, \quad i=\overline{1,n}, \quad j=\overline{1,m}, \tag{0.6}$$

де n – кількість постачальників (вантажовідправників); m – кількість споживачів (вантажоодержувачів); a_i – обмеження за пропозицією (сумарна кількість вантажу, що постачається i-м постачальником); b_j – обмеження за попитом (сумарна потреба в кількості вантажу j-го одержувача); c_{ij} – ваговий коефіцієнт, що визначає внесок у формування цільової функції одиниці вантажу, перевезеного від i-го постачальника до j-го одержувача; x_{ij} – об'єм кореспонденції (кількість вантажу, що транспортується) *між* i-м постачальником і j-*м* одержувачем.

4. *Розв'язання задачі перевезень.* Етап полягає у визначенні змінних (невідомих) x_{ij}, $i=\overline{1,n}$, $j=\overline{1,m}$, у математичній моделі області припустимих рішень (0.4) – (0.6), що забезпечують мінімум функції витрат на перевезення вантажу (0.3). Навіть при малому числі постачальників і споживачів $(n,m<10)$ вирішення транспортної задачі методом прямого перебору всіх можливих варіантів пов'язане зі значними труднощами. При числі постачальників і споживачів $(n,m\ge 10)$ оптимальне вирішення

задачі методом прямого перебору проблематично навіть із використанням сучасної обчислювальної техніки.

5. *Визначення маршрутів перевезень*. На цьому етапі визначається, за якими маршрутами – маятниковим або розвізними (кільцевими) – буде поставлятися вантаж від постачальників до споживачів за результатами вирішення транспортної задачі (0.3) – (0.6). У випадку, коли постачальнику та споживачам відповідає схема типу «один-до-багатьох», розв'язання цієї задачі зводиться до розв'язання задачі комівояжера [4].

6. *Моделювання часу перевезень*. Моделюється час поставки товару кожному зі споживачів. Визначення часу, протягом якого транспортні засоби перевозять вантаж, здійснюється за формулою

$$T_н = \sum_{j}^{A} t_{nj} + \sum_{k}^{B} t_{грk} + \sum_{j}^{C} t_{pj} + \sum_{k}^{B} t_{xk} + \sum_{j}^{A} \eta_j + \sum_{i}^{C} \psi_i , \qquad (0.8)$$

де t_{nj} – час навантаження транспортного засобу *j*-м постачальником, включаючи час чекання навантаження; *A* – кількість постачальників, що забезпечують завантаження транспортного засобу, $A \le n$; $t_{грk}$ – час руху транспортного засобу з вантажем на *k*-й ділянці; *B* – кількість неоднорідних ділянок на маршруті; t_{xk} – час руху транспортного засобу без вантажу на k-й ділянці; t_{pi} – час розвантаження вантажу для *i*-го споживача, включаючи час чекання розвантаження; *C* – кількість споживачів, яким розвозиться вантаж, $C \le m$; η_j – випадкова величина, що враховує обідні (технологічні) перерви *j*-го постачальника; ψ_i – випадкова складова, що враховує обідні (технологічні) перерви *i*-го споживача.

Для міжнародного перевезення загальний час знаходження автомобіля в рейсі (на маршруті) визначається за формулою

$$T_o = \sum_{i=1}^{A} t_{i,i+1} + \sum_{j=1}^{B} \tau_j + \sum_{k=1}^{C} \Theta_k + \sum_{l=1}^{D} \varphi_l + \sum_{m=1}^{E} \psi_m + \sum_{n=1}^{F} \eta_n , \qquad (0.9)$$

де $t_{i,i+1}$ – час руху між i-м і $(i+1)$-м пунктами маршруту; A – кількість ділянок руху автомобіля на маршруті; τ_j – час оформлення митних документів у j-му пункті; B – кількість пунктів митного оформлення на маршруті; Θ_k – час навантаження, розвантаження і складування в k-му пункті; C – кількість пунктів навантаження-розвантаження; φ_l – випадкова складова, що відображує збільшення часу рейсу для проведення ремонтно-профілактичних робіт і інших причин простою транспортного засобу; ψ_m – випадкова складова, що відображує обмеження, що зв'язані з ЕСТР; η_n – випадкова складова, що відображує заборони на рухи великовантажних автомобілів; D, E, F – число випадків простою автомобіля з врахуванням останніх трьох зазначених чинників відповідно.

Розрахувавши час рейсу, можна визначити час прибуття до кожного зі споживачів.

7. *Аналіз виконання вимог і умов перевезень*. Виконується перевірка відповідності очікуваних строків доставки строкам, обговореним із споживачем. У випадку, якщо умови споживача не виконуються, здійснюється коректування маршрутів, вантажопідйомності транспортного засобу, розкладу роботи складів і таке інше. У випадку, коли коректування приводить до зміни математичної моделі процесу перевезення, етапи 4 – 7 виконуються наново.

8. *Завантаження транспортного засобу та безпосередня реалізації перевезень*. Фактично, етап є критерієм усього процесу перевезення. Нормальне завершення етапу свідчить про правильну організацію транспортування вантажу. Якщо виконання етапу виявляє які-небудь небажані відхилення в запланованого

процесу перевезення, то їх оперативне усунення здійснюється не в межах цього алгоритму. Усі виявлені відхилення, як небажані, так і позитивні, повинні бути враховані при моделюванні наступних перевезень.

Недоліком наведеного алгоритму є його трудомісткість і орієнтованість на невелику кількість постачальників і споживачів [4, 6]. Крім того, нераціональним є побудова маршрутів без урахування вантажопідйомності автомобіля, що здійснює перевезення вантажу. Зокрема, для групи «постачальник – споживачі», можливо, прийдеться вирішувати задачу розвезення, а не задачу комівояжера. Рішення транспортної задачі з урахуванням вантажопідйомності доступних транспортних засобів може не збігатися з рішенням, отриманим без урахування вантажопідйомності транспортного засобу. У статті [6] автори пропонують наступні доповнення:

– після формування бази даних, що описують транспортну мережу, пропонується локалізувати склади та клієнтів за територіальною ознакою;

– процес маршрутизації сполучати з вибором транспортних засобів відповідної вантажопідйомності.

Наведений алгоритм планування реальних вантажних автомобільних перевезень, як правило, має лінійну структуру – всі етапи виконуються послідовно один за іншим. Але, у загальному випадку прагнення врахувати всі чинники, що впливають на процес транспортування вантажу, всі альтернативні маршрути і способи доставки, приводить до порушення лінійності. При цьому схема може мати як лінійні, так і розгалужені та циклічні фрагменти.

Для підвищення ефективності процесу транспортних перевезень, у т.ч. і автомобільних, застосовують формальний (науково обґрунтований) або евристичний підхід. Останній полягає в використанні експертів, що володіють досвідом організації подібних процесів [6]. Такий підхід не гарантує найбільш вдалого використання транспортних ресурсів або отримання максимального економічного ефекту від процесу транспортування. Зовсім інші результати забезпечує науковий підхід. Саме він використовує математичні моделі та методи пошуку оптимальних параметрів транспортного процесу, що гарантують максимальний економічний ефект. Крім наукового та евристичного підходу, може бути застосований змішаний підхід, коли організація деяких етапів вантажних перевезень будуються на евристичному підході, деякі – на формальному. Кожний із цих підходів може бути визначальним при організації транспортного процесу або бути присутнім поряд з іншими.

Дослідження процесу транспортних перевезень ведеться як за кордоном, так і в країнах СНД, зокрема, в Росії й Україні. Відповідно до доповіді міністра Росії в 2008 році, собівартість автомобільних перевезень у Росії в півтора рази перевищує собівартість у розвинутих країнах світу. У більшості випадків витрати можуть бути значно зменшені за рахунок оптимізації й автоматизації процесу постачання [7]. Відповідно до даних української конференції [8] в Україні при організації транспортних перевезень частіше усього використовується евристичний підхід, тобто планування процесу перевезень експертами на підставі напрацьованого досвіду. Іноді застосовуються комерційні пакети програм, однак відповідно до того ж джерела обидва підходи забезпечують приблизно однакову ефективність. Зіставляючи ці висновки з висновками, зробленими в [7] по Росії, стає очевидним, що комерційні пакети програм застосовуються неефективно, або вони не забезпечують належної оптимальності організації процесу перевезень.

У практиці організації транспортних перевезень найбільш широке комерційне застосування знайшли наступні програмні пакети:

1. Системи *GPS*-контролю. Це програмне забезпечення призначене винятково для контролю співробітників і персоналій, які здійснюють перевезення. Такий контроль дозволяє уникати витрат, до яких ведуть зловживання та недисциплінованість

співробітників. Системи *GPS*-контролю не призначені для оптимізації транспортного процесу.

2. Системи *TMS*-управління. Це програмне забезпечення являє собою універсальний пакет, що дозволяє виконувати наступні операції:

- управління замовленнями на перевезення;
- планування та формування маршрутів;
- обслуговування нестандартних транспортних подій;
- можливість взаємодії з різного типу мобільними пристроями;
- розрахунок вартості транспортних послуг;
- обслуговування договорів із зовнішніми транспортними компаніями;
- установлювання прейскурантів транспортних послуг;
- вибір алгоритмів розрахунку транспортних послуг;
- ведення статистики та аналізу даних за транспортною логістикою.

3. Географічні інформаційні системи (ГІС). Програмне забезпечення даного типу дозволяє складати маршрути в автоматичному режимі (із використанням убудованої в програму карти місцевості з транспортними комунікаціями) і визначати час проходження маршрутів. Останні версії удосконалених систем *TMS*-управління використовують ГІС для складання маршрутів.

Наведені програмні засоби дозволяють вирішувати задачі організації перевезення вантажу оптимальним способом тільки для транспортних мереж невеликої вимірності. У випадку транспортних мереж великої вимірності існуюче програмне забезпечення, у т.ч. і наведене, або не дозволяє одержати рішення, або не забезпечує його оптимальність. В останньому випадку приблизне рішення може стати причиною зайвих транспортних витрат.

Додатковим недоліком існуючого програмного забезпечення є використання критеріїв оптимізації з обмеженою гнучкістю, що значно знижує їх прикладне значення.

На підставі вивченої літератури можна дістати висновку, що організації транспортного процесу з погляду логістики є актуальною задачею. Проте системне вирішення задачі ефективної організації автомобільних перевезень вимагає розробки методів, що дозволяють оптимальним чином вирішувати окремі її підзадачі. Деякі із підзадач у літературних джерелах детально розглянуті, інші – мало вивчені. Методи вирішення багатьох підзадач були розроблені без врахування системності.

Таким чином, вивчення можливостей поліпшення організації транспортного процесу, безумовно, залишається актуальною проблемою. Успішне її вирішення може бути досягнуто тільки при комплексному розгляді всіх етапів і складових транспортного процесу із залученням додаткового контингенту вчених і дослідників.

РОЗДІЛ 1

МАТЕМАТИЧНІ МОДЕЛІ В ЗАДАЧАХ ОРГАНІЗАЦІЇ ТА ПЛАНУВАННЯ ВАНТАЖНИХ ПЕРЕВЕЗЕНЬ

Серед усіх етапів планування транспортних перевезень етапи «моделювання процесу перевезень» і «розв'язання транспортної задачі» залишаються центральними. Пояснюється це тим, що моделювання процесу перевезення з наступним визначенням оптимального розподілу наявних об'ємів вантажу у постачальників між одержувачами дозволяє не тільки повною мірою забезпечити вимоги останніх, але і зробити це з мінімальними витратами. Іншими словами, такий підхід дозволяє «на папері» дістати прибуток, що втрачається при евристичному або інтуїтивному плануванні перевезень. При цьому аналітичні методи пошуку раціональних варіантів дозволяють уникнути процесу прямого перебору варіантів.

На рис. 1.1 наведено схему, що класифікує задачі організації перевезень. Ці задачі за змістовною постановкою безпосередньо стосуються процесу транспортування вантажів або пасажирів. За математичними моделями вони належать до класу детермінованих задач лінійного неперервного або дискретного програмування.

Як бачимо, всі задачі за співвідношенням кількості постачальників (пунктів відправлення) та кількості замовників (пунктів призначення) підрозділяються на чотири типи:

- «один-до-одного»;
- «один-до-багатьох»;
- «багато-до-одного»;
- «багато-до-багатьох».

Рис. 1.1 – Класифікація задач з організації перевезень

Всі задачі організації перевезень можна розподілити також за їх призначенням. На рис. 1.1 для кожного типу задач, що розподілені за ознакою постачальник/замовник, перелічено найбільш характерні класи задач, що відрізняються один від одного призначенням. Так, для задач типу «один-до-одного» це:

- вибір транспортних засобів та розподіл вантажу;
- формування парку машин та їх розподіл.

Задачі типу «один-до-багатьох» складаються з класів:

- розподіл транспортних засобів за витратами часу;
- розподіл транспортних засобів за обсягом перевезень;
- розподіл транспортних засобів за витратами коштів;
- розподіл транспортних засобів із фіксованими доплатами;
- формування парку машин та їх розподіл;
- розвезення вантажу;
- вибір маршруту.

Тип задач «багато-до-одного» має майже такі ж самі класи, що й тип «один-до-багатьох»:

- планування випуску продукції;
- розподіл транспортних засобів за витратами часу;
- розподіл транспортних засобів за обсягом перевезень;
- розподіл транспортних засобів за витратами коштів;
- розподіл транспортних засобів із фіксованими доплатами;
- формування парку машин та їх розподіл;
- звезення вантажу;
- вибір маршруту.

Найбільш поширеним типом задач організації перевезень є останній тип «багато-до-багатьох». Саме до цього типу належить класична транспортна задача (0.3) – (0.6) та її різновиди. Він об'єднує такі класи задач:

- класична транспортна задача;
- транспортна задача з неперервною відкритою математичною моделлю;
- транспортна задача з дискретною математичною моделлю;
- перевезення вантажу за два етапи;
- перевезення вантажу декількох видів за два етапи;
- перевезення вантажу декількох видів за два етапи за замовленнями споживачів;
- закриття заводу;
- розиграш кубку;
- розбиття транспортної мережі на райони.

Перелік класів за призначенням налічує 25 найменувань. Тут треба зауважити, що цей перелік не претендує на вичерпаність. Він містить тільки найбільш відомі класи. Головне інше – кожний клас можна розподілити ще на підкласи за видом математичної моделі задачі. Кожний клас задач має декілька різновидів, що відрізняються однин від одного або виглядом цільової функції, або виглядом обмежень на змінні або тим та іншим одночасно. Загальну кількість підкласів важко визначити. Практика перевезень постійно пропонує нові математичні моделі, тому зосереджувати увагу на кожній моделі нема сенсу. Проте надалі принаймні одна з математичних моделей для кожного класу задач буде докладно розглянута.

При подальшому розгляді математичних моделей незалежно від класу задач, до якого вони належать, буде наведено відповідні загальні змістовні та математичні постановки задач, конкретні приклади задач, математичні та цифрові моделі в інформаційному середовищі *Microsoft Excel* (за умовами прикладів), розв'язання прикладів за допомогою програмного засобу

«Пошук рішення» системи *Microsoft Excel* і аналіз результатів. Такий розгляд може послужити прекрасним довідником для побудови математичних моделей або посібником для розв’язання реальних виробничих транспортних задач невеликої вимірності. На жаль, вимірність задач обмежується як обчислювальними ресурсами комп'ютера, на якому інстальоване середовище *Microsoft Excel*, так і алгоритмічними особливостями програми «Пошук рішення».

Проблема вимірності стосується всіх класів задач з організації перевезень. В подальшому при розгляді різних класів задач увага на проблемі вимірності зосереджуватися не буде. Виключення складають тільки задачі розподілу транспортної мережі на райони та задача маршрутизації (комівояжера), яким присвячені окремі розділи монографії та розроблені спеціальні методи розв’язання. Саме вимірність цих задач та алгоритми їх розв’язання будуть у центрі уваги.

Наступні підрозділи безпосередньо присвячені математичним та цифровим моделям задач, що за типом і класом відповідають класифікатору на рис. 1.1.

1.1. Задачі зі схемою перевезень типу «один-до-одного»

Задачі типу «один-до-одного» складаються з двох класів:

- вибір транспортних засобів і розподіл вантажу;
- формування парку машин та їх розподіл.

Розглянемо послідовно обидва класи.

1.1.1. Задача про вибір транспортних засобів і розподіл вантажу

Змістовна постанова задачі

Нехай вантажний флот має у своєму складі судна n типів.

Кількість суден типу j дорівнює q_j, а витрати при використанні одного судна типу j у плановому періоді складає c_j, j=1, 2, …, n. Кожне судно має вантажні ємності m типів (трюми, палуби, танки і т.п.). Вантажопідйомність i-ї ємності на судні типу j дорівнює d_{ij}, i=1, 2, …, m. Перевезенню підлягають p видів вантажу. Вантаж виду k є в кількості a_k, k=1,2,…, p. Потрібно вибрати найбільш економічний комплекс засобів для перевезення вантажу й об'єми вантажу, що будуть перевозитися кожною ємністю.

Математична постанова задачі

Позначимо через x_j кількість суден j-го типу, j=1, 2,…, n, що виділяються для перевезення, а через z_{ik} – кількість вантажу виду k, що завантажується в ємність i, k=1, 2, …, p. Тоді математична модель задачі вибору засобів постачання вантажу має вигляд:

$$y = \sum_{j=1}^{n} c_j x_j \to \min_{x_j \in \Omega}, \tag{1.1}$$

$$\Omega: \quad \sum_{j=1}^{n} d_{ij} x_j - \sum_{k=1}^{p} z_{ik} \ge 0, \quad i = \overline{1, m}, \tag{1.2}$$

$$\sum_{i=1}^{m} z_{ik} = a_k, \quad k = \overline{1, p}, \tag{1.3}$$

$$0 \le x_j \le q_j, \quad j = \overline{1, n}, \tag{1.4}$$

$$x_j = \text{int}, \quad j = \overline{1, n}, \tag{1.5}$$

$$z_{ik} \ge 0, \quad i = \overline{1, m}, \quad k = \overline{1, p}. \tag{1.6}$$

Тут цільова функція (1.1) визначає сумарні витрати на перевезення вантажу. Система нерівностей (1.2) обмежує загальну кількість вантажу, що завантажується в ємності кожного типу i, вантажопідйомністю цих ємностей для кожного із суден. Система нерівностей (1.3) вимагає вивезення вантажу у повному обсязі. Система двосторонньої обмеженості змінних x_j (1.4) разом із системами (1.5) і (1.6) визначають простір можливих рішень відповідно до умов задачі та фізичної суті змінних.

Приклад задачі про вибір транспортних засобів і розподіл вантажу

Нехай вантажний флот має у своєму складі судна чотирьох типів. Кількість суден j-го типу (j = 1, 2, 3, 4) відповідно дорівнює 15, 20, 30, 25. Витрати при використанні одного судна j-го типу в плановому періоді складають відповідно 6, 5, 7, 4 *од*. Кожне судно має вантажні ємності трьох типів (трюми, палуби, танки). Вантажопідйомність d_{ij} кожної i–ї ємності (i=1, 2, 3) на судні типу j визначається матрицею $\mathbf{D} = \begin{bmatrix} 4 & 3 & 3 & 2 \\ 3 & 2 & 3 & 4 \\ 2 & 2 & 2 & 3 \end{bmatrix}$. Перевезенню підлягає вантаж двох видів ($p = 2$). Вантаж виду k (k=1, 2) є наявним відповідно в кількості 220 і 130. Потрібно вибрати найбільш економічний комплекс засобів перевезення вантажу й об'єми вантажу у кожній ємності.

Математична модель задачі про вибір транспортних засобів і розподіл вантажу

Математична модель задачі при використанні позначень, що прийняті для загальної моделі цієї задачі (1.1) – (1.6), буде мати вигляд:

$$y = 6x_1 + 5x_2 + 7x_3 + 4x_4 \to \min_{x_j \in \Omega},$$

$$\mathbf{\Omega}: \quad f_1 = 4x_1 + 3x_2 + 3x_3 + 2x_4 - (z_{11} + z_{12}) \geq 0 \ ,$$

$$f_2 = 3x_1 + 2x_2 + 3x_3 + 4x_4 - (z_{21} + z_{22}) \geq 0 \ ,$$

$$f_3 = 2x_1 + 2x_2 + 2x_3 + 3x_4 - (z_{31} + z_{32}) \geq 0 \ ,$$

$$f_4 = z_{11} + z_{21} + z_{31} = 220 \ ,$$

$$f_5 = z_{12} + z_{22} + z_{32} = 130 ,$$

$$0 \leq x_1 \leq 15 \ ,$$

$$0 \leq x_2 \leq 20 \ ,$$

$$0 \leq x_3 \leq 30 \ ,$$

$$0 \leq x_4 \leq 25 \ ,$$

$$x_j = \text{int}, \ \ j = \overline{1,4} \ ,$$

$$z_{ik} \geq 0, \ \ i = \overline{1,3}; \ \ k = 1,2 \ .$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для задачі про вибір засобів доставки вантажу в умовах прикладу може бути таким:

- клітинка B6:E6 – для шуканих змінних задачі x_j, $j = \overline{1,4}$;
- клітинки L5:M7 – для шуканих змінних задачі z_{ik}, $i = \overline{1,3}$, $k = 1,2$ **;**
- клітинки B4:E4 – для заданих кількостей суден d_j кожного типу, $j = \overline{1,4}$;
- клітинки B9:E9 – для заданих витрат c_j на використання одного судна *j*–го типу, $j = \overline{1,4}$ **;**

- клітинки B11:E11 – для проміжних результатів із завантаженими формулами для обчислення добутку $c_j x_j$, $j = \overline{1,4}$;

- клітинки G5:J7 – для указівки ємностей транспортних засобів d_{ij}, $i = \overline{1,3}$, $j = \overline{1,4}$;

- клітинки L9:M9 – для функцій f_{3+k}, $k = \overline{1,2}$, із завантаженими формулами для обчислення суми $z_{1k} + z_{2k} + z_{3k}$;

- клітинки L11:M11 – для задання об'ємів вантажу a_k, $k = 1,2$, що підлягає вивезенню;

- клітинки G14:J16 – для проміжних результатів із завантаженими формулами для обчислення добутку $d_{ij} x_j$, $i = \overline{1,3}$, $j = \overline{1,4}$;

- клітинки L14:L16 – для проміжних результатів із завантаженими формулами для обчислення суми $d_{i1}x_1 + d_{i2}x_2 + d_{i3}x_3 + d_{i4}x_4$, $i = \overline{1,3}$;

- клітинки N14:N16 – для проміжних результатів із завантаженими формулами для обчислення суми $z_{1k} + z_{2k}$, $k = 1,2$;

- клітинки P14:P16 – для остаточного обчислення функцій f_i, $i = \overline{1,3}$, з відповідними завантаженими формулами;

- клітинка C13 – для значення цільової функції *y* із завантаженою формулою для обчислення суми $\sum_{j=1}^{4} c_j x_j$.

Всі клітинки, що призначені для цифрової моделі, повинні мати *числовий* формат із числом десяткових знаків, рівним 2. Виключення складають клітинки з цілочисловими даними. Це – клітинки B6:E6 для шуканих змінних задачі. Вони повинні мати числовий формат із числом десяткових знаків, рівним 0.

Після завантаження в електронну таблицю всіх необхідних констант і формул для обчислення проміжних і кінцевого результатів треба обов'язково виконати необхідні установки даних у діалоговому вікні команди *Сервіс/Пошук_ рішення*. В умовах прикладу установки визначаються в такий спосіб:

- для цільової клітинки – **$C$13**, рівної **мінімальному значенню;**
- для клітинок із змінними – **$B$6:$E$6**;**$L$5:$M$7**;
- для обмежень: **$B$6:$E$6 <= $B$4:$E$4**;
  $B$6:$E$6 = ціле;
  $B$6:$E$6 >= 0;
  $L$5:$M$7 >= 0;
  $L$9:$M$9 = $L$11:$M$11;
  $P$14:$P$16 >= 0.

Вигляд діалогового вікна з необхідними установками показаний на (рис. 1.2).

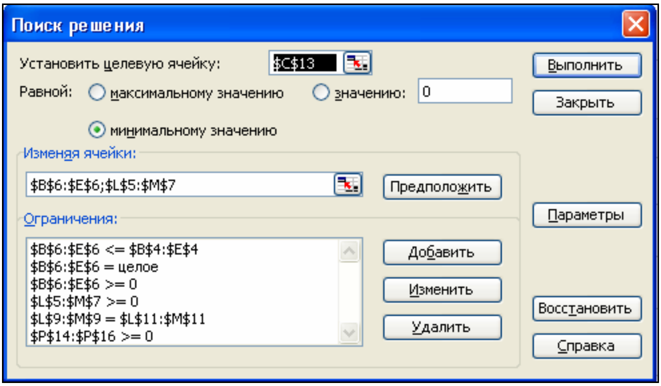


Рис. 1.2 – Діалогове вікно команди *Сервіс/Пошук_ рішення* після установки початкових даних

На рис.1.3 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_ рішення*.

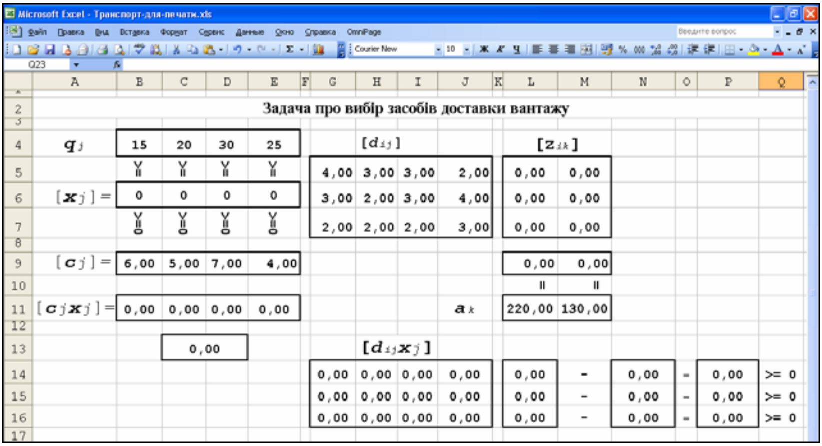


Рис. 1.3 – Екран з початковими установками

Вигляд екрана після виконання команди *Сервіс/Пошук_ рі-шення* показаний на рис. 1.4.

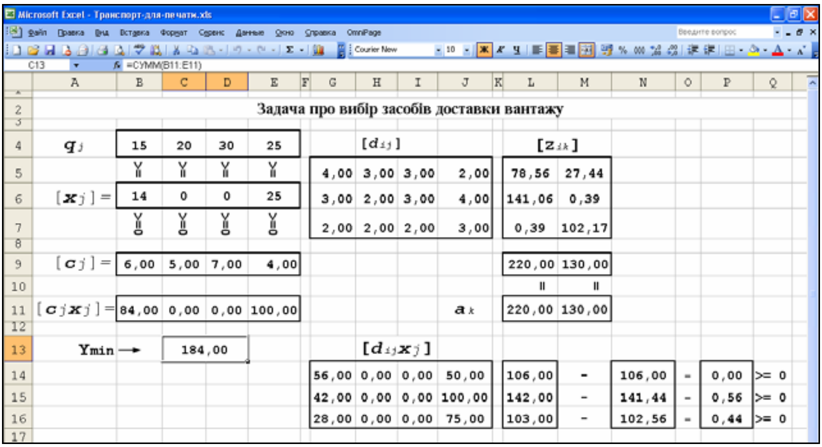


Рис. 1.4 – Екран з проміжними і кінцевими результатами

Оптимальним рішенням задача є вектор-рядок $\mathbf{X}^{*T} = \begin{bmatrix} 14 & 0 & 0 & 25 \end{bmatrix}$ і матриця $\mathbf{Z}^* = \begin{bmatrix} 78{,}56 & 27{,}44 \\ 141{,}06 & 0{,}39 \\ 0{,}39 & 102{,}17 \end{bmatrix}$, які забезпечують мінімальне значення цільової функції $y^* = 184$ *ум.од.*

Таким чином, для найбільш економічного перевезення всього вантажу треба використати 14 суден 1-го типу та 25 суден 4-го типу. При цьому 78,56 *од*. вантажу 1-го виду та 27,44 *од*. вантажу 2-го виду треба завантажувати в ємності суден 1-го типу; 141,06 *од*. вантажу 1-го виду та 0,39 *од*. вантажу 2-го виду – в ємності 2-го типу; 0,39 *од*. вантажу 1-го виду та 102,17 *од*. вантажу 2-го виду – в ємності 3-го типу.

1.1.2. Задача про формування парку машин та їх розподіл

Змістовна постанова задачі

Маємо *n* типів транспортних засобів, що пристосовані до перевезення *m* видів залізобетонних конструкцій. Кожний *j*-й тип транспортного засобу спроможний одночасно перевозити залізобетонні конструкції *i*–го виду в кількості a_{ij}, $i = \overline{1,m}$ $j = \overline{1,n}$. З житлово-будівельного комбінату треба перевезти на будівельний майданчик конструкції в кількості b_i, $i = \overline{1,m}$. Відома собівартість c_{ij} вантажно-розвантажувальних робіт однієї конструкції *i*–го виду при перевезенні на транспортному засобі *j*-го типу, а також витрати d_j на оренду та утримання транспортного засобу *j*-го типу. Потрібно визначити оптимальний машинний парк і розподіл транспортних засобів між конструкціями, щоб забезпечити їх доставку на будівельний майданчик з мінімальною сумарною вартістю.

Математична модель задачі

Позначимо змінні:

x_j – кількість транспортних засобів *j*–го типу; $j = \overline{1,n}$;

y_{ij} – кількість транспортних засобів *j*–го типу, що залучаються до перевезення залізобетонних конструкцій *i*–го виду.

Тоді задача про формування парку машин зводиться до задачі цілочислового лінійного програмування:

$$\sum_{j=1}^{n} d_j x_j + \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij} y_{ij} \to \min_{x_j, y_{ij} \in \Omega}, \qquad (1.7)$$

$$\Omega: \quad \sum_{j=1}^{n} a_{ij} y_{ij} \ge b_i, \quad i = \overline{1,m}, \qquad (1.8)$$

$$x_j - \sum_{i=1}^{m} y_{ij} \ge 0_i, \quad j = \overline{1,n}, \qquad (1.9)$$

$$x_j \ge 0, \quad x_j = \text{int}, \quad j = \overline{1,n}, \qquad (1.10)$$

$$y_{ij} \ge 0, \quad y_{ij} = \text{int}, \quad j = \overline{1,n}, \quad i = \overline{1,m}. \qquad (1.11)$$

Тут цільова функція (1.7) визначає сумарні витрати на перевезення залізобетонних конструкцій та оренду транспортних засобів. Система обмежень у вигляді нерівностей (1.8) показує, що транспортних засобів повинно вистачати для перевезення залізобетонних конструкцій усіх видів. Система нерівностей

(1.9) вимагає, щоб кількість розподілених транспортних засобів кожного типу не перевищувала їх загальної кількості. Системи обмежень (1.10) і (1.11) визначають простір можливих рішень відповідно до умови задачі та фізичної природи змінних.

Приклад задачі про формування парку машин та їх розподіл

Є 5 типів транспортних засобів і 3 види залізобетонних конструкцій, відповідно в кількостях 70, 120 і 150. Відомі місткості a_{ij} транспортних засобів відносно кожного виду конструкцій та собівартість c_{ij} робіт з навантаження та розвантаження *i*–го виду конструкцій відносно *j*–го типу транспортних засобів відповідно у вигляді матриць **A** та **C,** $i = \overline{1,3},\ j = \overline{1,5}$ **:**

$$\mathbf{A} = \left[a_{ij}\right] = \begin{bmatrix} 2 & 3 & 3 & 1 & 2 \\ 3 & 2 & 4 & 2 & 3 \\ 4 & 2 & 3 & 2 & 4 \end{bmatrix};$$

$$\mathbf{C} = \left[c_{ij}\right] = \begin{bmatrix} 20 & 30 & 20 & 40 & 30 \\ 30 & 40 & 20 & 30 & 30 \\ 40 & 20 & 20 & 10 & 20 \end{bmatrix}.$$

Оренда машин кожного типу складає відповідно 40, 50, 30, 45, та 35 *тис. ум. од.* Треба визначити оптимальний машинний парк та розподіл транспортних засобів між залізобетонними конструкціями для забезпечення їх доставки до місця будівельних робіт з мінімальною сумарною вартістю.

Математична модель задачі про формування парку машин та їх розподіл

Математична модель задачі при використанні позначень, що прийняті для загальної моделі цієї задачі (1.7) – (1.11), буде мати вигляд:

$$z = 40x_1 + 50x_2 + 30x_3 + 45x_4 + 35x_5 + 20y_{11} + 30y_{21} + 40y_{31} +$$
$$+ 30y_{12} + 40y_{22} + 20y_{32} + 20y_{13} + 20y_{23} + 20y_{33} +$$
$$+ 40y_{14} + 30y_{24} + 10y_{34} + 30y_{15} + 30y_{25} + 20y_{35} \to \min_{x_{ij} \in \Omega},$$

$$\Omega: \quad f_1 = 2y_{11} + 3y_{12} + 3y_{13} + y_{14} + 2y_{15} \ge 70 \ ,$$

$$f_2 = 3y_{21} + 2y_{22} + 4y_{23} + 2y_{24} + 3y_{25} \ge 120 \ ,$$

$$f_3 = 4y_{31} + 2y_{32} + 3y_{33} + 2y_{34} + 4y_{35} \ge 150 \ ,$$

$$f_4 = x_1 - (y_{11} + y_{21} + y_{31}) \ge 0 \ ,$$

$$f_5 = x_2 - (y_{12} + y_{22} + y_{32}) \ge 0,$$

$$f_6 = x_3 - (y_{13} + y_{23} + y_{33}) \ge 0 \ ,$$

$$f_7 = x_4 - (y_{14} + y_{24} + y_{34}) \ge 0 \ ,$$

$$f_8 = x_5 - (y_{15} + y_{25} + y_{35}) \ge 0 \ ,$$

$$x_j \ge 0, \quad x_j = \text{int}, \quad j = \overline{1,5} \ ,$$

$$y_{ij} \ge 0, \quad y_{ij} = \text{int}, \quad i = \overline{1,3}; \quad j = \overline{1,5} \,.$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для задачі про формування парку машин та їх розподіл в умовах прикладу може бути таким:

- клітинка B10:F10 – для шуканих змінних задачі x_j, $j = \overline{1,5}$;
- клітинки B12:F14 – для шуканих змінних задачі y_{ij}, $i = \overline{1,3}$, $j = \overline{1,5}$;
- клітинки B4:F6 – для задання місткості a_{ij} транспортних засобів відносно кожного типу будівельних конструкцій $i = \overline{1,3}$, $j = \overline{1,5}$;
- клітинки H4:L6 – для задання собівартості c_{ij} робіт з навантаження та розвантаження кожного виду конструкцій відносно кожного типу транспортних засобів c_{ij}, $i = \overline{1,3}$, $j = \overline{1,5}$;
- клітинки B8:F8 – для задання витрат d_j на оренду та утримання транспортного засобу j-го типу, $j = \overline{1,5}$;
- клітинки P4:P6 – для задання кількості виготовлених конструкцій b_i кожного виду, $i = \overline{1,5}$;
- клітинки N4:N6 – для функцій f_i, $i = \overline{1,3}$, із завантаженими формулами для обчислення суми $a_{i1}y_{i1} + a_{i2}y_{i2} + a_{i3}y_{i3} + {}$ $+ a_{i4}y_{i4} + a_{i5}y_{i5}$;
- клітинки B16:F16 – для функцій f_{3+j}, $j = \overline{1,5}$, із завантаженими формулами для обчислення виразу $x_j - (y_{1j} + y_{2j} + y_{3j})$;

- клітинка H10 – для проміжного результату із завантаженою формулою для обчислення суми $\sum_{j=1}^{5} d_j x_j$;

- клітинка J10 – для проміжного результату із завантаженою формулою для обчислення подвійної суми $\sum_{i=1}^{3}\sum_{j=1}^{5} c_{ij} y_{ij}$;

- клітинка L10 – для значення цільової функції z із завантаженою формулою для її обчислення.

Всі клітинки цифрової моделі, що призначені для грошових величин, повинні мати *числовий* формат із числом десяткових знаків, рівним 2. Це клітинки: H4:L6; B8:F8; H10; J10; L10. Інші клітинки призначені для цілочислових даних. Вони повинні мати числовий формат із числом десяткових знаків, рівним 0.

Установки даних в діалоговому вікні команди *Сервіс/Пошук рішення*:

цільова клітинка – **$L$10**;

тип екстремуму – **мінімум**;

клітинки зі змінними – **$B$10:$F$10;$B$12:$F$14**;

обмеження: **$B$10:$F$10 = ціле**;
$B$10:$F$10 >= 0;
$B$12:$F$14 >= 0;
$B$12:$F$14 = ціле;
$B$16:$F$16 >= 0;
$N$4:$N$6 >= $P$4:$P$6.

На рис.1.5 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.

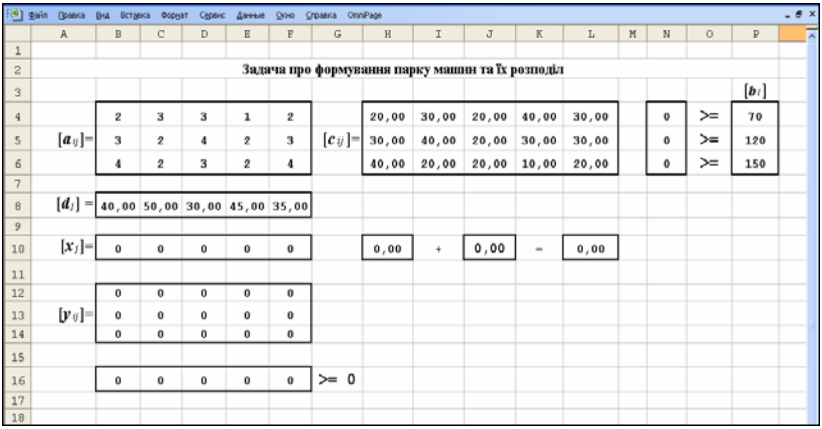


Рис. 1.5 – Екран з початковими установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.6.

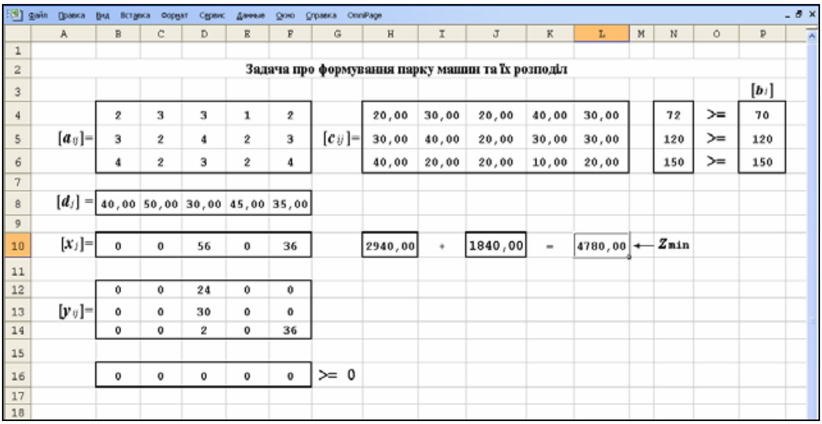


Рис. 1.6 – Екран з проміжними і кінцевими результатами

Оптимальний розв'язок задачі складається з вектор-рядка

$$\mathbf{X}^{*T} = \begin{bmatrix} 0 & 0 & 56 & 0 & 36 \end{bmatrix} \text{ і матриці } \mathbf{Y}^{*} = \begin{bmatrix} 0 & 0 & 24 & 0 & 0 \\ 0 & 0 & 30 & 0 & 0 \\ 0 & 0 & 2 & 0 & 36 \end{bmatrix}.$$

Компоненти вектору $\mathbf{X}^{*}$ свідчать про те, що за умовами задачі оптимальний варіант парку машин повинен складатися з 56 машин 3-го типу та 36 машин 5-го типу. При цьому всі машини 5-го типу призначаються для перевезення будівельних конструкцій 3-го виду, а 2 конструкції 3-го типу та всі конструкції 1-го, так саме як і 2-го, транспортуються машинами 3-го типу. Такий вибір і розподіл парку машин забезпечує мінімальні витрати на перевезення будівельного вантажу в розмірі $z^{*} = 4780$ *ум. од.*

1.2. Задачі організації перевезень типу «один-до-багатьох»

Задачі типу «один-до-багатьох» складаються з семи класів:

- розподіл транспортних засобів за витратами часу;
- розподіл транспортних засобів за обсягом перевезень;
- розподіл транспортних засобів за витратами коштів;
- розподіл транспортних засобів із фіксованими доплатами;
- формування парку машин та їх розподіл;
- розвезення вантажу;
- вибір маршруту.

Задачі цього типу в практиці вантажних перевезень зустрічаються досить часто. Вони притаманні транспортним і виробничим підприємствам, коли вантаж треба з одного місця (підприємства, складу і т.п.) перевезти до декількох замовників.

1.2.1. Задача про розподіл транспортних засобів за витратами часу

Змістовна постановка задачі

Для перевезення піску з одного піщаного кар'єру на n будівельних майданчиків використовуються транспортні засоби m типів. Продуктивність i-ї транспортної одиниці на j-му маршруті (на j-й майданчик) становить t_{ij} тонн за добу, $i = \overline{1,m}$, $j = \overline{1,n}$. Резерв корисного часу i-го транспортного засобу обмежений величиною a_i діб, $i = \overline{1,m}$. Мінімальний об'єм вантажу, що перевозиться на j–му маршруті становить b_j тонн, $j = \overline{1,n}$. Визначити, які саме транспортні засоби, на яких маршрутах і на протязі якого терміну (скількох діб) потрібно використовувати, щоб забезпечити перевезення всього вантажу при мінімальних витратах загального резерву корисного часу. Вважається, що кожний транспортний засіб на протязі однієї доби може обслуговувати тільки один маршрут, тобто кількість часу вимірюється цілим числом діб.

Математична постановка задачі

Позначимо через x_{ij} кількість діб, на протязі яких транспортний засіб i-го типу використовується на j-му маршруті. Тоді математична модель задачі про розподіл транспортних засобів з мінімальним витратами загального часу буде мати вигляд:

$$y = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \to \min_{x_{ij} \in \Omega}, \qquad (1.12)$$

$$\Omega: \quad \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = \overline{1,m}, \qquad (1.13)$$

$$\sum_{i=1}^{m} t_{ij} x_{ij} \geq b_j, \quad j = \overline{1,n} \;, \tag{1.14}$$

$$x_{ij} \geq 0; \quad i = \overline{1,m}; \quad j = \overline{1,n} \;; \tag{1. 15}$$

$$x_{ij} = \text{int}; \quad i = \overline{1,m}; \quad j = \overline{1,n} \,. \tag{1.16}$$

Тут цільова функція (1.12) визначає загальну витрату часу на перевезення всього вантажу за всіма маршрутами. Система нерівностей (1.13) обмежує сумарні витрати часу транспортним засобом кожного типу. Система нерівностей (1.14) забезпечує виконання заданих обсягів перевезень на всіх маршрутах. Вирази (1.15) і (1.16) обмежують простір припустимих рішень відповідно до суті змінних.

Приклад задачі про розподіл транспортних засобів за витратами часу

Для перевезення вантажу на трьох транспортних маршрутах використовуються транспортні засоби трьох типів. Продуктивність (у *тоннах за добу*) кожної транспортної одиниці на кожному маршруті визначається матрицею $\mathbf{T} = \begin{bmatrix} 8 & 14 & 11 \\ 6 & 15 & 13 \\ 12 & 12 & 4 \end{bmatrix}$, резерви корисного часу транспортних засобів становлять відповідно 280, 320 і 340 діб. Мінімальний обсяг вантажу, що підлягає перевезенню на кожному з маршрутів відповідно становить 3000, 5400, 3300 *тонн*. Кожна транспортна одиниця на протязі однієї доби може використовуватися тільки на одному з трьох можливих маршрутів. Визначити, які саме транспортні засоби, на яких маршрутах і на протязі якого строку потрібно використовувати, щоб забезпечити перевезення всього вантажу при мінімальних загальних витратах часу.

Математична модель задачі про розподіл транспортних засобів за витратами часу в умовах прикладу

Математична модель задачі при використанні позначень, прийнятих для загальної моделі (1.12) – (1.16), буде мати вигляд:

$$y = x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + x_{31} + x_{32} + x_{33} \to \min_{x_{ij} \in \Omega},$$

$$\Omega: \quad f_1 = x_{11} + x_{12} + x_{13} \le 280,$$

$$f_2 = x_{21} + x_{22} + x_{23} \le 320,$$

$$f_3 = x_{31} + x_{32} + x_{33} \le 340,$$

$$f_4 = 8x_{11} + 6x_{21} + 12x_{31} \ge 3000,$$

$$f_5 = 14x_{12} + 15x_{22} + 12x_{32} \ge 5400,$$

$$f_6 = 11x_{13} + 13x_{23} + 4x_{33} \ge 3300,$$

$$x_{ij} \ge 0, \quad i = \overline{1,3}, \quad j = \overline{1,3},$$

$$x_{ij} = \text{int}; \quad i = \overline{1,3}, \quad j = \overline{1,3}.$$

Цифрова модель і розв’язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для задачі про розподіл транспортних засобів з мінімальними витратами резерву корисного часу в умовах прикладу може бути таким:

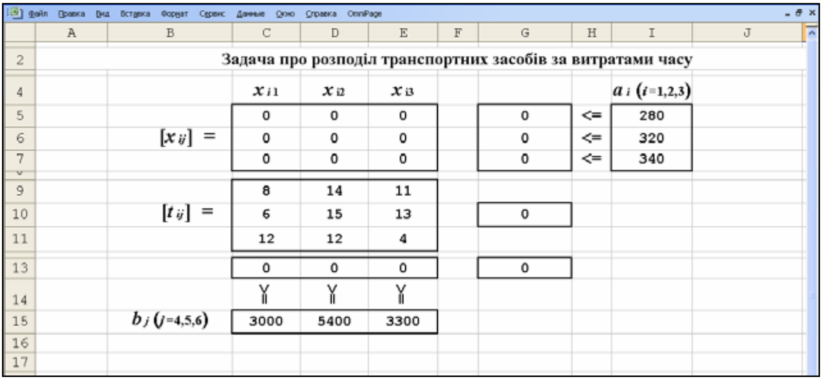
- клітинки C5:E7 – для шуканих значень змінних задачі x_j, $j = \overline{1,3}$;
- клітинки C9:E11 – для задання продуктивності транспортних засобів t_{ij}, $i = \overline{1,3}$, $j = \overline{1,3}$;
- клітинки G5:G7 – для функцій f_i, $i = \overline{1,3}$, із завантаженими формулами для обчислення суми $x_{i1} + x_{i2} + x_{i3}$;
- клітинки C13:E13 – для функцій f_{3+j}, $j = \overline{1,3}$, із завантаженими формулами для обчислення суми $t_{1j}x_{1j} + t_{2j}x_{2j} + t_{3j}x_{3j}$;
- клітинки I5:I7 – для заданих резервів корисного часу a_i, $i = \overline{1,3}$, кожного транспортного засобу;
- клітинки C15:E15 – для заданих обсягів перевезень b_j, $j = \overline{1,3}$, для кожного маршруту;
- необов'язкова клітинка G13 – для контролю за загальним обсягом перевезень, який не може бути меншим за $\sum_{j=1}^{3} b_i = 11700$;
- клітинка G10 – для значення цільової функції *y* із завантаженою формулою для підрахунку подвійної суми $\sum_{i=1}^{3}\sum_{j=1}^{3} x_{ij}$.

Всі клітинки, що призначені для цифровій моделі, повинні мати *числовий* формат із числом десяткових знаків, рівним 0.

Після завантаження в електронну таблицю всіх необхідних констант і формул для обчислення проміжних і кінцевого результатів треба обов'язково виконати необхідні установки даних у діалоговому вікні команди *Сервіс/Пошук_ рішення*. В умовах прикладу установки визначаються в такий спосіб:

- для цільової клітинки – **$G$10**, рівної **мінімальному значенню;**
- для клітинок зі змінними – **$C$5:$E$7**;
- для обмежень: **$C$13:$E$13 <= $C$15:$E$15**;
  $C$5:$E$7 = ціле;
  $C$5:$E$7 >= 0;
  $G$5:$G$7 >= $I$5:$I$7.

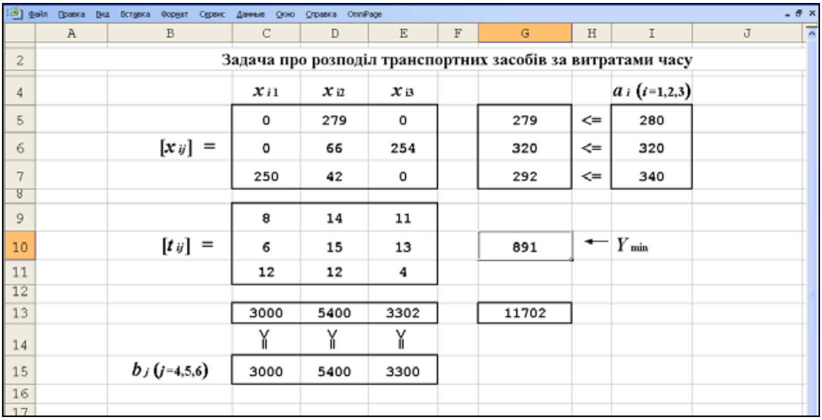
На рис. 1.7 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.



	A	B	C	D	E	F	G	H	I	J
2			Задача про розподіл транспортних засобів за витратами часу							
4			x_{i1}	x_{i2}	x_{i3}				a_i $(i=1,2,3)$	
5			0	0	0		0	<=	280	
6		$[x_{ij}]$ =	0	0	0		0	<=	320	
7			0	0	0		0	<=	340	
9			8	14	11					
10		$[t_{ij}]$ =	6	15	13		0			
11			12	12	4					
13			0	0	0		0			
14			≤	≤	≤					
15		b_j $(j=4,5,6)$	3000	5400	3300					
16										
17										

Рис. 1.7 – Екран з початковими установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.8.



Задача про розподіл транспортних засобів за витратами часу

	x_{i1}	x_{i2}	x_{i3}			a_i (i=1,2,3)
	0	279	0	279	<=	280
$[x_{ij}]$ =	0	66	254	320	<=	320
	250	42	0	292	<=	340
	8	14	11			
$[t_{ij}]$ =	6	15	13	891	←	Y_{min}
	12	12	4			
	3000	5400	3302	11702		
	≥	≥	≥			
b_j (j=4,5,6)	3000	5400	3300			

Рис. 1.8 – Екрана з проміжними та кінцевими результатами

Оптимальним рішенням задачі є матриця значень змінних $\mathbf{X}^* = \begin{bmatrix} 0 & 279 & 0 \\ 0 & 66 & 254 \\ 250 & 42 & 0 \end{bmatrix}$, що забезпечує мінімальне значення цільової функції $y^* = 891\,д$. При такому рішенні фонд корисного часу буде вичерпаний тільки для другого транспортного засобу. При цьому 66 діб він буде працювати на 2-му маршруті та 254 доби – на третьому. Перший транспортний засіб буде обслуговувати тільки 2-й маршрут. При цьому у нього залишиться резервною 1 доба. Третій транспортний засіб 250 діб буде працювати на 1-му маршруті та 42 доби – на 2-му. Невикористаними залишаться 48 діб. Потреби в вантажу будуть повністю виконані. Більш того, на 3-му маршруті можна додатково перевезти ще 2 *т*. Залишковий резерв часу, що мають перший та третій транспортний засіб доцільно використати на інших роботах, що принесе додатковий прибуток транспортному підприємству.

1.2.2. Задача про розподіл транспортних засобів за обсягом перевезень

Змістовна постановка задачі

Для перевезення піску з одного піщаного кар'єру на *n* будівельних майданчиків використовуються транспортні засоби *m* типів. Продуктивність *i*-ї транспортної одиниці на *j*-му маршруті (на *j*-й майданчик*)* становить t_{ij} тонн за добу, $i = \overline{1,m},\ j = \overline{1,n}$. Резерв корисного часу *i*-го транспортного засобу обмежений величиною a_i діб, $i = \overline{1,m}$. Мінімальний об'єми вантажу, що перевозиться на *j*–му маршруті становить b_j *тонн*, $j = \overline{1,n}$. Визначити, які саме транспортні засоби, на яких маршрутах і на протязі якого терміну (скількох діб) потрібно використовувати, щоб забезпечити перевезення максимально можливого об'єму вантажу. Вважається, що кожен транспортний засіб на протязі однієї доби може обслуговувати тільки один маршрут.

Примітка. Ця задача відрізняється від попередньої тільки своєю математичною моделлю, а саме – цільовою функцією. Якщо в попередній задачі за мету малося мінімальне використання резерву корисного часу, то зараз – максимальне використання всіх транспортних засобів, тобто перевезення максимального загального об'єму вантажу.

Математична постановка задачі

Позначимо, як і раніше, через x_{ij} кількість діб, на протязі яких транспортний засіб *i*-го типу використовується на *j*-му маршруті. Тоді математична модель задачі про розподіл транспортних засобів з максимальним обсягом перевезень набуває вигляду:

$$y = \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} x_{ij} \to \max_{x_{ij} \in \Omega}\ , \qquad (1.17)$$

$$\mathbf{\Omega}: \quad \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = \overline{1,m} , \tag{1.18}$$

$$\sum_{i=1}^{m} t_{ij} x_{ij} \ge b_j, \quad j = \overline{1,n} , \tag{1.19}$$

$$x_{ij} \ge 0; \quad i = \overline{1,m}; \quad j = \overline{1,n} ; \tag{1. 20}$$

$$x_{ij} = \text{int}; \quad i = \overline{1,m}; \quad j = \overline{1,n} . \tag{1.21}$$

Тут цільова функція (1.17) визначає загальний об’єм перевезень за всіма маршрутами. Область припустимих рішень Ω така ж сама, як в попередній задачі.

Приклад задачі про розподіл транспортних засобів за обсягом перевезень

Для перевезення вантажу на трьох транспортних маршрутах використовуються транспортні засоби трьох типів. Продуктивність (у *тоннах за добу*) кожної транспортної одиниці на кожному маршруті визначається матрицею $\mathbf{T} = \begin{bmatrix} 8 & 14 & 11 \\ 6 & 15 & 13 \\ 12 & 12 & 4 \end{bmatrix}$, резерви корисного часу транспортних засобів становлять відповідно 380, 320 і 340 діб. Мінімальний обсяг вантажу, що підлягає перевезенню на кожному з маршрутів відповідно становить 3000, 5400, 3300 *тонн*. Кожна транспортна одиниця на протязі однієї доби може використовуватися тільки на одному з трьох можливих маршрутів. Визначити, які саме транспортні засоби, на яких маршрутах і на протязі якого строку потрібно використовувати, щоб забезпечити максимальний загальний об’єм вантажу.

Математична модель задачі про розподіл транспортних засобів за обсягом перевезень в умовах прикладу

Математична модель задачі при використанні позначень, прийнятих для загальної моделі (1.17) – (1.21), буде складатися с цільової функції

$$y = 8x_{11} + 14x_{12} + 11x_{13} + 6x_{21} + 15x_{22} + 13x_{23} +$$
$$+ 12x_{31} + 12x_{32} + 4x_{33} \to \max_{x_{ij} \in \Omega}$$

та області припустимих рішень Ω, що вже була визначена у попередньому прикладі.

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Формат і розподіл за призначенням всіх клітинок електронної таблиці для задачі про розподіл транспортних засобів з максимальним обсягом перевезень є таким самим, як у попередньому прикладі. Виключення складають тільки клітинки:

- G10 – зараз вона відіграє роль необов'язкової клітинки, що використовується для порівняння з аналогічною клітинкою попереднього прикладу;
- G13 – для значення цільової функції y із завантаженою формулою для підрахунку подвійної суми $\sum_{i=1}^{3}\sum_{j=1}^{3} t_{ij} x_{ij}$.

Установки в діалоговому вікні команди *Сервіс/Пошук_ рішення* в умовах прикладу визначаються в такий спосіб:

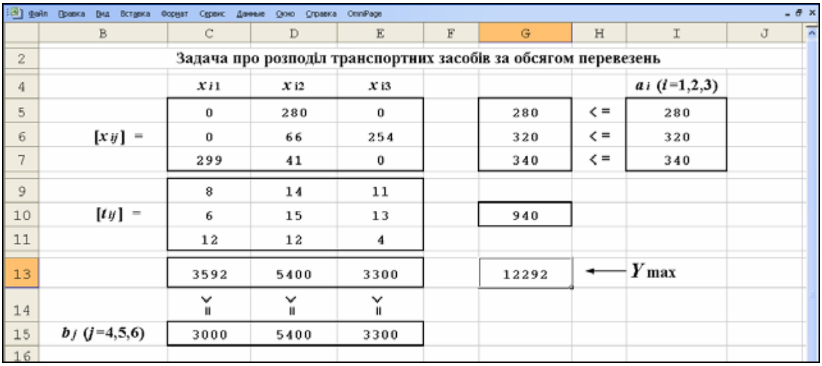
- для цільової клітинки – **$G$13**, рівної **максимальному значенню;**
- для клітинок зі змінними – **$C$5:$E$7**;
- для обмежень: **$C$13:$E$13 <= $C$15:$E$15**;

$C$5:$E$7 = ціле;

$C$5:$E$7 >= 0;

$G$5:$G$7 >= $I$5:$I$7.

Вигляд екрана до виконання команди *Сервіс/Пошук_рішення* повністю співпадає з тим, що зображений на рис. 1.7. Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.9.


Файл Правка Вид Вставка Формат Сервис Данные Окно Справка OmniPage

	B	C	D	E	F	G	H	I	J
2	Задача про розподіл транспортних засобів за обсягом перевезень								
4		x_{i1}	x_{i2}	x_{i3}				a_i (i=1,2,3)	
5		0	280	0		280	<=	280	
6	[x_{ij}] =	0	66	254		320	<=	320	
7		299	41	0		340	<=	340	
9		8	14	11					
10	[t_{ij}] =	6	15	13		940			
11		12	12	4					
13		3592	5400	3300		12292	←	Y max	
14		>=	>=	>=					
15	b_j (j=4,5,6)	3000	5400	3300					
16									

Рис. 1.9 – Екран з проміжними та кінцевими результатами

Оптимальним рішенням задачі є матриця значень змінних

$$\mathbf{X}^* = \begin{bmatrix} 0 & 280 & 0 \\ 0 & 66 & 254 \\ 299 & 41 & 0 \end{bmatrix},$$

що забезпечує максимальне значення цільової функції $y^* = 12\,292$ *т*. При такому рішенні фонд корисного часу буде вичерпаний повністю (про що свідчить клітинка G10). Об'єм вантажу в порівнянні з попереднім прикладом значно збільшився: був 11702 *т* (див. клітинку G13 на рис. 1.8), а став 12 292 *т*.

1.2.3. Задача про розподіл транспортних засобів за грошовими витратами

Змістовна постановка задачі

Нехай є *n* транспортних ліній, за якими щоденно розвозять продукцію з хлібозаводу. На *j*–й лінії необхідно виконати b_j рейсів, $j = \overline{1,n}$. У наявності є транспортні одиниці m типів. Резерви корисного часу транспортних одиниць типу *i* становлять a_i, $i = \overline{1,m}$. На виконання транспортною одиницею типу *i* рейсу на лінії *j* потрібен час t_{ij}, а грошові витрати на один рейс становлять c_{ij}. Потрібно визначити найбільш економічне розміщення транспортних одиниць за всіма лініями, тобто розміщення з мінімальними грошовими витратами на виконання всіх рейсів.

Математична постановка задачі

Позначимо через x_{ij} кількість рейсів, що транспортна одиниця *i* повинна виконати на лінії *j*. Тоді математична модель задачі про розподіл транспортних засобів за витратами коштів набуває вигляду:

$$y = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min_{x_{ij}\in\Omega}\ , \qquad (1.22)$$

$$\Omega: \quad \sum_{j=1}^{n} t_{ij}x_{ij} \le a_i, \quad i = \overline{1,m}\ , \qquad (1.23)$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1,n}\ , \qquad (1.24)$$

$$x_{ij} \geq 0; \quad i=\overline{1,m}; \quad j=\overline{1,n}\ ; \tag{1.25}$$

$$x_{ij} = \text{int}; \quad i=\overline{1,m}; \quad j=\overline{1,n}\,. \tag{1.26}$$

Тут цільова функція (1.22) відповідає сумарним витратам на виконання всіх рейсів за всіма лініями. Система нерівностей (1.23) обмежує сумарні витрати часу транспортними одиницями кожного i–го типу. Система рівностей (1.24) вимагає виконання потрібної за умови задачі кількості рейсів на кожній j-й лінії. Вирази (1.25) і (1.26) обмежують простір припустимих рішень відповідно до суті змінних.

Приклад задачі про розподіл транспортних засобів за грошовими витратами

Нехай є три транспортні лінії, на котрих необхідно виконати відповідно 11, 7, 9 рейсів. У наявності є транспортні одиниці трьох типів. Резерви корисного часу транспортних одиниць кожного типу відповідно становлять 100, 130, 250. Часові та грошові витрати на виконання транспортною одиницею кожного типу на кожній з трьох ліній задаються відповідно матрицями **T** і **C:** $\mathbf{T}=\begin{bmatrix}4&3&3\\2&2&3\\2&2&2\end{bmatrix}$, $\mathbf{C}=\begin{bmatrix}5&5&4\\3&3&5\\4&3&4\end{bmatrix}$**.** Потрібно визначити найбільш економічне розміщення транспортних одиниць за всіма лініями.

Математична модель задачі про розподіл транспортних засобів за грошовими витратами в умовах прикладу

Математична модель задачі в умовах прикладу при використанні позначень, прийнятих для загальної моделі (1.22) – (1.26), набуває вигляду:

$$y = 5x_{11} + 5x_{12} + 4x_{13} + 3x_{21} + 3x_{22} + 5x_{23} +$$

$$+ 4x_{31} + 3x_{32} + 4x_{33} \to \min_{x_{ij} \in \Omega} ,$$

$$\Omega: \quad f_1 = 4x_{11} + 3x_{12} + 3x_{13} \le 100 ,$$

$$f_2 = 2x_{21} + 2x_{22} + 3x_{23} \le 130 ,$$

$$f_3 = 2x_{31} + 2x_{32} + 2x_{33} \le 250 ,$$

$$f_4 = x_{11} + x_{21} + x_{31} = 11 ,$$

$$f_5 = x_{12} + x_{22} + x_{32} = 7 ,$$

$$f_6 = x_{13} + x_{23} + x_{33} = 9 ,$$

$$x_{ij} \ge 0, \quad i = \overline{1,3}, \quad j = \overline{1,3} ,$$

$$x_{ij} = \text{int}; \quad i = \overline{1,3}, \quad j = \overline{1,3} .$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для задачі про розподіл транспортних засобів за грошовими витратами в умовах прикладу може бути таким:

- клітинки B5:D7 – для шуканих значень змінних задачі x_{ij}, $i = \overline{1,3}$, $j = \overline{1,3}$;
- клітинки B9:D9 – для функцій f_{3+j}, $j = \overline{1,3}$, із завантаженими формулами для обчислення суми $x_{1j} + x_{2j} + x_{3j}$, $j = \overline{1,3}$;

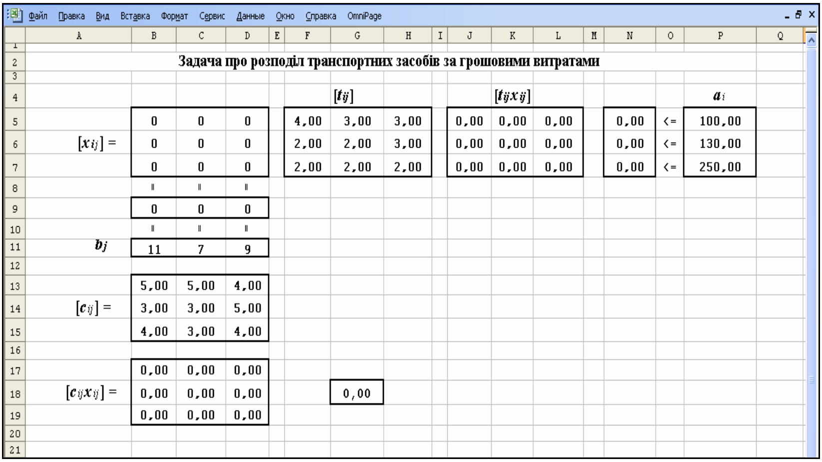
- клітинки B11:D11 – для задання кількості рейсів b_j, $j = \overline{1,3}$, на кожній транспортній лінії;
- клітинки B13:D15 – для елементів тарифної матриці **C**;
- клітинки B17: D19 – для проміжних результатів із завантаженими формулами для обчислення добутку $c_{ij} x_{ij}$, $i = \overline{1,3}$, $j = \overline{1,3}$;
- клітинки F5:H7 – для елементів строкової матриці **T**;
- клітинки J5:L7 – для проміжних результатів із завантаженими формулам для обчислення добутку $t_{ij} x_{ij}$, $i = \overline{1,3}$, $j = \overline{1,3}$;
- клітинки N5:N7 – для функцій f_i, $i = \overline{1,3}$, із завантаженими формулами для обчислення суми $t_{i1} x_{i1} + t_{i2} x_{i2} + t_{i3} x_{i3}$, $i = \overline{1,3}$;
- клітинки P5:P7 – для резервів корисного часу кожного типу транспортного засобу a_i, $i = \overline{1,3}$;
- клітинка G18 – для значення цільової функції y із завантаженою формулою для обчислення подвійної суми $\sum_{i=1}^{3} \sum_{j=1}^{3} c_{ij} x_{ij}$.

Всі клітинки електронної таблиці, що задіяні для цифрової моделі, повинні мати *числовий* формат з двома десятковими знаками. Виключенням є клітинки із цілими даними. Це клітинки B5:D7 (для шуканих змінних задачі), B9:D9 (для сум шуканих змінних) і B11:D11 (для кількості рейсів на кожній лінії).

Установки в діалоговому вікні команди *Сервіс/Пошук_ рішення* в умовах прикладу визначаються в такий спосіб:

- для цільової клітинки – **$G$18**, рівної **мінімальному значенню;**
- для клітинок зі змінними – **$B$5:$D$7;**
- для обмежень: **$B$5:$D$7 ціле;**
  $B$5:$D$7 >= 0;
  $B$9:$D$9 = $B$11:$D$11;
  $N$5:$N$7 < = $P$5:$P$7.

На рис. 1.10 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2		Задача про розподіл транспортних засобів за грошовими витратами															
3																	
4							[t_{ij}]				[$t_{ij}x_{ij}$]					a_i	
5		0	0	0		4,00	3,00	3,00		0,00	0,00	0,00		0,00	<=	100,00	
6	[x_{ij}] =	0	0	0		2,00	2,00	3,00		0,00	0,00	0,00		0,00	<=	130,00	
7		0	0	0		2,00	2,00	2,00		0,00	0,00	0,00		0,00	<=	250,00	
8		‖	‖	‖													
9		0	0	0													
10		‖	‖	‖													
11	b_j	11	7	9													
12																	
13		5,00	5,00	4,00													
14	[c_{ij}] =	3,00	3,00	5,00													
15		4,00	3,00	4,00													
16																	
17		0,00	0,00	0,00													
18	[$c_{ij}x_{ij}$] =	0,00	0,00	0,00			0,00										
19		0,00	0,00	0,00													
20																	
21																	

Рис. 1.10 – Екран із початковими установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.11.

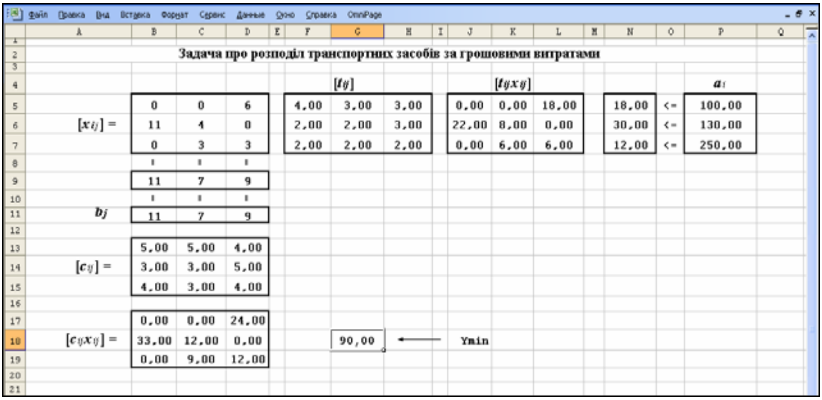


Рис. 1.11 – Екран з проміжними і кінцевими результатами

Оптимальним рішенням задача є матриця значень змінних $\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 6 \\ 11 & 4 & 0 \\ 0 & 3 & 3 \end{bmatrix}$, що забезпечує мінімум цільової функції $y^* = 90$ *ум.од.*

1.2.4. Задача про розподіл транспортних засобів із фіксованими доплатами

Змістовна постановка задачі

Нехай є n транспортних ліній, за якими щоденно розвозять продукцію з хлібозаводу. На j–й лінії необхідно виконати b_j рейсів, $j = \overline{1, n}$. У наявності є транспортні одиниці m типів. Резерви корисного часу транспортних одиниць типу i становлять a_i, $i = \overline{1, m}$. На виконання транспортною одиницею типу i

рейсу на лінії *j* потрібен час t_{ij}, а грошові витрати на один рейс становлять c_{ij}. Потрібно визначити найбільш економічне розміщення транспортних одиниць по лініях, тобто мінімальні грошові витрати на виконання всіх рейсів. Крім того, на відміну від попередньої задачі про розподіл транспортних засобів за грошовими витратами, випуск транспортної одиниці типу *i* на лінію *j* пов'язаний із підготовчими роботами, що вимагають витрат часу в розмірі t_{ij}^{+}. Цей час не залежить від числа рейсів, що повинен виконати транспортний засіб. Грошові витрати на проведення цих підготовчих робіт становлять c_{ij}^{+}. Потрібно визначити найбільш економічне розміщення транспортних одиниць за всіма лініями.

Математична постановка задачі

Як і в попередній задачі, за x_{ij} позначимо кількість рейсів, що транспортна одиниця *i* повинна виконати на лінії *j*. Тоді математична модель задачі про розподіл транспортних засобів із фіксованими доплатами набуває вигляду:

$$y = \sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}(x_{ij}) \to \min_{x_{ij}\in\Omega}, \qquad (1.27)$$

$$\Omega: \quad \sum_{j=1}^{n} h_{ij}(x_{ij}) \le a_i, \quad i=\overline{1,m}, \qquad (1.28)$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j=\overline{1,n}, \qquad (1.29)$$

$$x_{ij} \ge 0, \quad i=\overline{1,m}, \quad j=\overline{1,n}, \qquad (1.30)$$

$$x_{ij} = \text{int}, \quad i=\overline{1,m}, \quad j=\overline{1,n}. \qquad (1.31)$$

де

$$s_{ij}(x_{ij}) = \begin{cases} 0, & x_{ij} = 0, \\ c_{ij}x_{ij} + c_{ij}^{+}, & x_{ij} > 0, \end{cases} \tag{1.32}$$

$$h_{ij}(x_{ij}) = \begin{cases} 0, & x_{ij} = 0, \\ t_{ij}x_{ij} + t_{ij}^{+}, & x_{ij} > 0. \end{cases} \tag{1.33}$$

Тут цільова функція (1.27) відповідає сумарним витратам на виконання всіх рейсів по всіх лініях з урахуванням фіксованих доплат. Система нерівностей (1.28) обмежує сумарні витрати часу транспортними одиницями кожного i–го типу також з урахуванням додаткових витрат на підготовчі роботи – витрати не повинні перевершувати резерву корисного часу. Система рівностей (1.29) моделює вимоги на загальну кількість рейсів на кожній лінії. Системи (1.30) і (1.31) обмежують припустимий простір змінних відповідно до сутті самих змінних.

Перервні функції (1.32) і (1.33) відповідно визначають витрати коштів та часу на виконання j-ю транспортною одиницею на i-й лінії з урахуванням разових витрат на підготовчі роботи.

Приклад задачі про розподіл транспортних засобів із фіксованими доплатами

Нехай є три транспортні лінії, на котрих необхідно виконати відповідно 11, 7, 9 рейсів. У наявності є транспортні одиниці трьох типів. Резерви корисного часу транспортних одиниць кожного типу відповідно складають 100, 130, 250. Витрати часу та коштів на виконання рейсі транспортною одиницею кожного типу на кожній з трьох ліній задаються відповідно матрицями: $\mathbf{T} = \begin{bmatrix} 4 & 3 & 3 \\ 2 & 2 & 3 \\ 2 & 2 & 2 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 5 & 5 & 4 \\ 3 & 3 & 5 \\ 4 & 3 & 4 \end{bmatrix}$. Час t_{ij}^{+}, пов'язаний із підготовчими роботами на випуск транспортної одиниці типу i на лінію j, і грошові витрати c_{ij} на проведення цих підготовчих робіт задаються відповідно матрицями: $\mathsf{T}^+ = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 3 \\ 1 & 3 & 2 \end{bmatrix}$ та $\mathbf{C}^+ = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 1 & 2 \\ 2 & 1 & 1 \end{bmatrix}$. Потрібно визначити найбільш економічне розміщення транспортних одиниць за всіма лініями.

Математична модель задачі про розподіл транспортних засобів з фіксованими доплатами в умовах прикладу

Математична модель задачі в умовах прикладу при використанні позначень, прийнятих для загальної моделі (1.27) – (1.33), набуває вигляду:

$$y = s_{11} + s_{12} + s_{13} + s_{21} + s_{22} + s_{23} + s_{31} + s_{32} + s_{33} \to \min_{x_{ij} \in \mathbf{\Omega}},$$

$$\mathbf{\Omega}: \quad f_1 = h_{11} + h_{12} + h_{13} \le 100,$$

$$f_2 = h_{21} + h_{22} + h_{23} \le 130,$$

$$f_3 = h_{31} + h_{32} + h_{33} \le 250,$$

$$f_4 = x_{11} + x_{21} + x_{31} = 11,$$

$$f_5 = x_{12} + x_{22} + x_{32} = 7,$$

$$f_6 = x_{13} + x_{23} + x_{33} = 9,$$

$$x_{ij} \ge 0, \quad i = \overline{1,3}, \quad j = \overline{1,3},$$

$$x_{ij} = \text{int}, \quad i = \overline{1,3}, \quad j = \overline{1,3} \ .$$

Тут

$$s_{11} = \begin{cases} 0, & x_{11} = 0, \\ 5x_{11} + 1, & x_{11} > 0, \end{cases} \qquad s_{12} = \begin{cases} 0, & x_{12} = 0, \\ 5x_{12} + 2, & x_{12} > 0, \end{cases}$$

$$s_{13} = \begin{cases} 0, & x_{13} = 0, \\ 4x_{13} + 2, & x_{13} > 0, \end{cases} \qquad s_{21} = \begin{cases} 0, & x_{21} = 0, \\ 3x_{21} + 1, & x_{21} > 0, \end{cases}$$

$$s_{22} = \begin{cases} 0, & x_{22} = 0, \\ 3x_{22} + 1, & x_{22} > 0, \end{cases} \qquad s_{23} = \begin{cases} 0, & x_{23} = 0, \\ 5x_{23} + 2, & x_{23} > 0, \end{cases}$$

$$s_{31} = \begin{cases} 0, & x_{31} = 0, \\ 4x_{31} + 2, & x_{31} > 0, \end{cases} \qquad s_{32} = \begin{cases} 0, & x_{32} = 0, \\ 3x_{32} + 1, & x_{32} > 0, \end{cases}$$

$$s_{33} = \begin{cases} 0, & x_{33} = 0, \\ 4x_{33} + 1, & x_{33} > 0, \end{cases} \qquad h_{11} = \begin{cases} 0, & x_{11} = 0, \\ 4x_{11} + 1, & x_{11} > 0, \end{cases}$$

$$h_{12} = \begin{cases} 0, & x_{12} = 0, \\ 3x_{12} + 2, & x_{12} > 0, \end{cases} \qquad h_{13} = \begin{cases} 0, & x_{13} = 0, \\ 3x_{13} + 1, & x_{13} > 0, \end{cases}$$

$$h_{21} = \begin{cases} 0, & x_{21} = 0, \\ 2x_{21} + 2, & x_{21} > 0, \end{cases} \qquad h_{22} = \begin{cases} 0, & x_{22} = 0, \\ 2x_{22} + 1, & x_{22} > 0, \end{cases}$$

$$h_{23} = \begin{cases} 0, & x_{23} = 0, \\ 3x_{23} + 3, & x_{23} > 0, \end{cases} \qquad h_{31} = \begin{cases} 0, & x_{31} = 0, \\ 2x_{31} + 1, & x_{31} > 0, \end{cases}$$

$$h_{32} = \begin{cases} 0, & x_{32} = 0, \\ 2x_{32} + 3, & x_{32} > 0, \end{cases} \qquad h_{33} = \begin{cases} 0, & x_{33} = 0, \\ 2x_{33} + 2, & x_{33} > 0. \end{cases}$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для задачі про розподіл транспортних засобів із фіксованими доплатами може бути таким:

- клітинки B5:D7 – для шуканих значень змінних задачі x_{ij}, $i = \overline{1,3}$, $j = \overline{1,3}$;
- клітинки B9:D9 – для функцій f_{3+j}, $j = \overline{1,3}$, із завантаженими формулами для обчислення сум $x_{1j} + x_{2j} + x_{3j}$, $j = \overline{1,3}$;
- клітинки B11:D11 – для задання необхідних кількостей рейсів b_j, $j = \overline{1,3}$, на транспортних лініях;
- клітинки B13:D15 – для елементів тарифної матриці **C**;
- клітинки F13:H15 – для елементів тарифної матриці $\mathbf{C}^+$ із фіксованими доплатами на підготовку транспортних засобів;
- клітинки F17:H19 – для проміжних результатів із завантаженими формулами для обчислення функцій s_{ij}, $i = \overline{1,3}$, $j = \overline{1,3}$;
- клітинки F5:H7 – для елементів матриці **T**, що визначає витрати часу на виконання рейсів;
- клітинки J5:L7 – для елементів матриці $\mathbf{T}^+$, що визначає витрати часу на підготовку транспортних засобів до рейсів;

- клітинки J9:L11 – для проміжних результатів із завантаженими формулами, що обчислюють функції h_{ij}, $i=\overline{1,3}$, $j=\overline{1,3}$;

- клітинки N9:N11 – для функцій f_i, $i=\overline{1,3}$, із завантаженими формулами для підрахунку сум $h_{i1}+h_{i2}+h_{i3}$, $i=\overline{1,3}$;

- клітинки P5:P7 – для резервів корисного часу кожного типу транспортного засобу a_i, $i=\overline{1,3}$, заданих умовою прикладу;

- осередок J18 – для значення цільової функції із завантаженою формулою для підрахунку подвійної суми $\sum_{i=1}^{3}\sum_{j=1}^{3} s_{ij}$.

Всі клітинки, що задіяні в цифровій моделі, повинні мати *числовий* формат із числом десяткових знаків, рівним 2. Виключення становлять клітинки з цілими даними. Це клітинки B5:D7 (для шуканих значень змінних задачі), B9:D9 (для сум шуканих значень змінних) і B11:D11 (для кількостей рейсів на лінії).

Установки в діалоговому вікні команди *Сервіс/Пошук_рішення* в умовах прикладу визначаються в такий спосіб:

- для цільового клітинки – **$J$18**, рівною **мінімальному значенню;**

- для клітинок із змінними – **$B$5:$D$7;**

- для обмежень: **$B$5:$D$7 ціле;**
  $B$5:$D$7 >= 0;
  $N$9:$N$11 <= $P$9:$P$11;
  $B$9:$D$9 = $B$11:$D$11.

На рис.1.12 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.

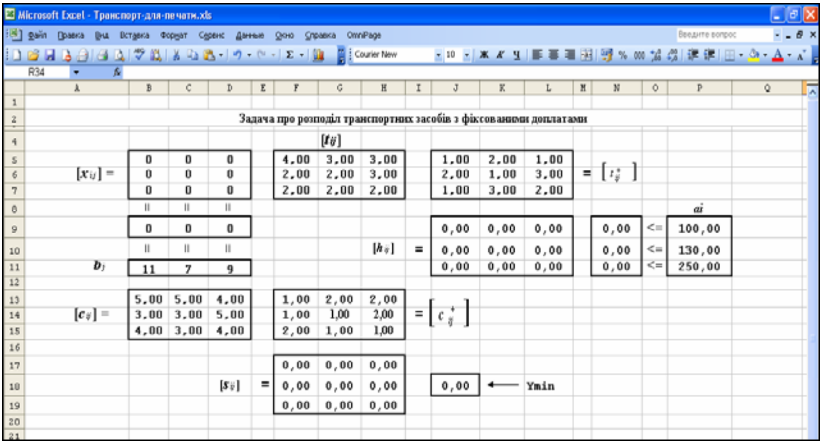


Рис. 1.12 – Екран із початковими установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.13.

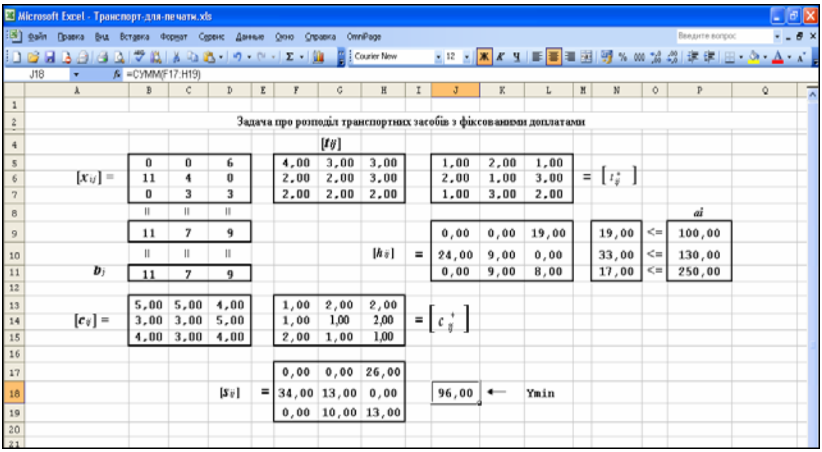


Рис. 1.13 – Екран із проміжними і кінцевими результатами

Оптимальним рішенням задачі є матриця значень змінних $\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 6 \\ 11 & 4 & 0 \\ 0 & 3 & 3 \end{bmatrix}$, що забезпечує мінімальне значення цільової функції $y^* = 96$ *ум. од.*

1.2.5. Задача про формування парку машин та їх розподіл

Змістовна постанова задачі

Маємо n типів агротехнічних машин і m видів аграрних робіт, що треба виконати в об'ємах b_i, $i = \overline{1,m}$. Вважається, що всі об'єми вимірюються в гектарах. Задана продуктивність j-ї машини на i-й роботі a_{ij}, $i = \overline{1,m}$, $j = \overline{1,n}$, а також собівартість c_{ij} обробки 1 *га* при виконанні i-ї роботи j-ю машиною. Відомо, що розмір оренди однієї машини j-го типу становить d_j *ум.од.*, $j = \overline{1,n}$. Треба визначити такий машинний парк до даного комплексу робіт і такий розподіл машин між роботами, щоб заплановане завдання виконати з мінімальними грошовими витратами.

Математична модель задачі

Позначимо змінні:

x_j – кількість транспортних засобів j–го типа, $j = \overline{1,n}$;

y_{ij} – кількість транспортних засобів j–го типу, що залучаються до виконання роботи i–го виду.

Тоді задача про формування парку машин зводиться до задачі цілочислового лінійного програмування:

$$z = \sum_{j=1}^{n} d_j x_j + \sum_{j=1}^{n}\sum_{i=1}^{m} c_{ij} y_{ij} \to \min_{x_j, y_{ij} \in \Omega}, \tag{1.34}$$

$$\Omega: \quad \sum_{j=1}^{n} a_{ij} y_{ij} = b_i, \quad i = \overline{1,m}, \tag{1.35}$$

$$x_j - \sum_{i=1}^{m} y_{ij} \ge 0_i, \quad j = \overline{1,n}, \tag{1.36}$$

$$x_j \ge 0, \quad x_j = \text{int}, \quad j = \overline{1,n}, \tag{1.37}$$

$$y_{ij} \ge 0, \quad y_{ij} = \text{int}, \quad j = \overline{1,n}, \quad i = \overline{1,m}. \tag{1.38}$$

Тут цільова функція (1.34) визначає сумарні витрати на виконання аграрних робіт та оренду агротехнічних засобів. Система обмежень у вигляді рівностей (1.35) показує, що роботи повинно виконати у повному обсязі. Система нерівностей (1.36) вимагає, щоб кількість розподілених агротехнічних машин кожного типу не перевершувала їх загальної кількості. Системи обмежень (1.37) і (1.38) визначають простір можливих рішень, що не суперечить умовам задачі.

Приклад задачі про формування парку машин та їх розподіл

Є 5 типів агротехнічних машин і 3 види аграрних робіт, що треба виконати відповідно на ділянках з площею 70, 120 і 150 *га*. Задані продуктивності кожної машини на кожній із робіт у вигляді матриці **A** та собівартість обробки

1 *га* на кожній роботі кожною з машин у вигляді матриці **С:**

$$\mathbf{A}=\left[a_{ij}\right]=\begin{bmatrix}7 & 8 & 5 & 9 & 6\\ 15 & 10 & 15 & 5 & 30\\ 12 & 15 & 20 & 5 & 30\end{bmatrix};\quad \mathbf{C}=\left[c_{ij}\right]=\begin{bmatrix}2 & 3 & 2 & 4 & 3\\ 3 & 4 & 2 & 3 & 3\\ 4 & 2 & 2 & 1 & 2\end{bmatrix}.$$

Оренда машин кожного типу складає відповідно 2,5; 4,0; 3,0; 3.2; 3,5 *тис.ум.од.* Треба визначити оптимальний машинний парк та розподіл машин між аграрними роботами таким чином, щоб забезпечити виконання віх обсягів робіт з мінімальним сумарними витратами.

Математична модель задачі про формування парку машин та їх розподіл в умовах прикладу

Математична модель задачі при використанні позначень, що прийняті для загальної моделі цієї задачі (1.34) – (1.38), буде мати вигляд:

$$\begin{aligned} z = {} & 2{,}5x_1 + 4x_2 + 3x_3 + 3{,}2x_4 + 3{,}5x_5 + 2y_{11} + 3y_{21} + 4y_{31} + \\ & + 3y_{12} + 4y_{22} + 2y_{32} + 2y_{13} + 2y_{23} + 2y_{33} + \\ & + 4y_{14} + 3y_{24} + y_{34} + 3y_{15} + 3y_{25} + 2y_{35} \to \min_{x_j,\, y_{ij} \in \Omega}, \end{aligned}$$

$$\begin{aligned} \Omega:\quad & f_1 = 7y_{11} + 8y_{12} + 5y_{13} + 9y_{14} + 6y_{15} = 70\ , \\ & f_2 = 15y_{21} + 10y_{22} + 15y_{23} + 5y_{24} + 30y_{25} = 120\ , \\ & f_3 = 20y_{31} + 15y_{32} + 20y_{33} + 5y_{34} + 30y_{35} = 150\ , \\ & f_4 = x_1 - \left(y_{11} + y_{21} + y_{31}\right) \ge 0\ , \\ & f_5 = x_2 - \left(y_{12} + y_{22} + y_{32}\right) \ge 0, \end{aligned}$$

$$f_6 = x_3 - \left(y_{13} + y_{23} + y_{33}\right) \geq 0 \ ,$$

$$f_7 = x_4 - \left(y_{14} + y_{24} + y_{34}\right) \geq 0 \ ,$$

$$f_8 = x_5 - \left(y_{15} + y_{25} + y_{35}\right) \geq 0 \ ,$$

$$x_j \geq 0, \quad x_j = \text{int}, \quad j = \overline{1{,}5} \ ,$$

$$y_{ij} \geq 0, \quad y_{ij} = \text{int}, \quad i = \overline{1{,}3}; \quad j = \overline{1{,}5} \, .$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для задачі про формування парку машин та їх розподіл в умовах цього прикладу повністю співпадає з тим, що був у аналогічній задачі в підрозділі 1.1.2.

Установки даних в діалоговому вікні команди *Сервіс/Пошук_рішення*:

цільова клітинка – **$N$10**;

тип екстремуму – **мінімум**;

клітинки зі змінними – **$B$10:$F$10;$B$12:$F$14**;

обмеження: **$B$10:$F$10 = ціле**;
$B$10:$F$10 >= 0;
$B$12:$F$14 >= 0;
$B$12:$F$14 = ціле;
$B$16:$F$16 >= 0;
$N$4:$N$6 = $P$4:$P$6.

На рис.1.14 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.

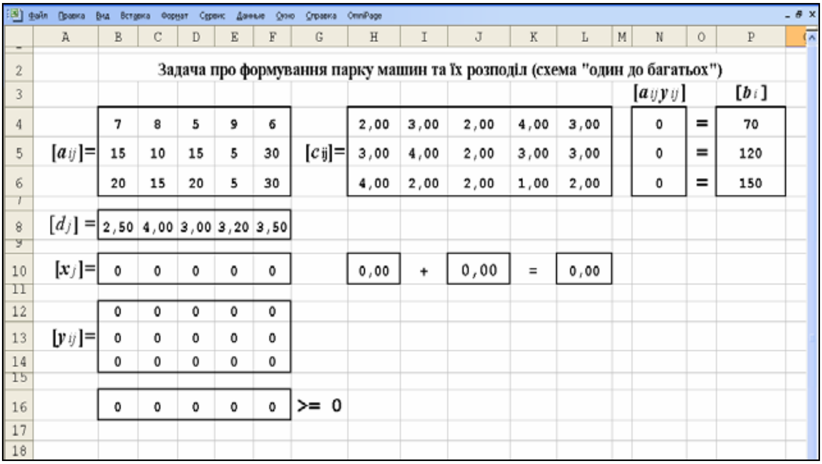


Рис. 1.14 – Екран з початковими установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.15.

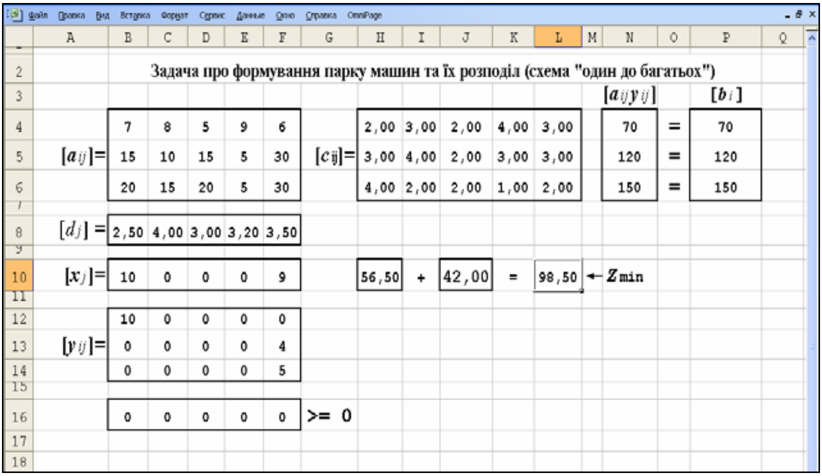


Рис. 1.15 – Екран з проміжними і кінцевими результатами

Оптимальний розв'язок задачі складається з вектор-рядка $\mathbf{X}^{*T} = \begin{bmatrix} 10 & 0 & 0 & 0 & 9 \end{bmatrix}$ і матриці $\mathbf{Y}^{*} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}$.

Компоненти вектору $\mathbf{X}^{*}$ свідчать про те, що за умовами задачі оптимальний варіант парку машин повинен складатися з 10 машин 1-го типу та 9 машин 5-го типу. При цьому всі машини 1-го типу призначаються для виконання аграрної роботи 1-го виду, а машину 5-го типу розподіляються між роботами 2-го та 3-го видів відповідно у кількостях 4 та 5. Такий вибір і розподіл парку машин забезпечує мінімальні витрати на виконання всіх аграрних робіт у розмірі $z^{*} = 98{,}5$ *тис. ум. од.*

Призначення розглянутої задачі таке ж саме, як і задачі з підрозділу 1.1.2. Обидві задачі мають схожі математичні та цифрові моделі. Відмінності лише становлять системи (1.8) та (1.35): одна є системою нерівностей (1.8) типу «більше або рівне», а друга – системою рівностей. За певними обставинами ці системи теж можуть бути однаковими. Проте задачі мають різні схеми перевезень. Перша задача, де перевезення відбуваються між житлово-будівельним комбінатом та будівельним майданчиком, тобто один постачальник, один замовник і один маршрут доставки, має схему перевезень «один-до-одного». Друга – схему «один-до- багатьох». Тут, безумовно, маршрути агротехнічних засобів до місць проведення робіт відрізняються один від одного. Ці відмінності можуть впливати на собівартості c_{ij} обробки 1 *га* при виконанні i-ї роботи j-ю машиною, або входити до оренди окремою складовою як фіксовані доплати.

На перший погляд наведений приклад не стосується безпосередньо перевезення вантажу. Але це не зовсім так. Якщо за аграрні роботи вважати, наприклад, внесення добрів, насіння, або гербіцидів, то одразу виникають питання з їх перевезенням на ділянки. При цьому одиниці виміру робіт можуть бути як гектари, так і тони.

1.2.6. Задача про розвезення вантажу

Змістовна постановка задачі

Деяка центральна база постачає товар (його можна вважати однорідним) на m складів. Розвезення товару на склади здійснюється однією вантажівкою, причому кожний склад одержує своє замовлення цілком за один прийом – вантажопідйомність вантажівки для цього достатня. Вантажівка може одночасно узяти вантаж, що відповідає не більш ніж k замовленням. Вантажівка може об'їжджати склади за r маршрутами. Один і тій самий склад може знаходитися на різних маршрутах.

Нехай для кожного складу відома функція витрат у залежності, наприклад, від розміру замовлення. Потрібно скласти графік розвезень товару, що забезпечить виконання замовлень всіх клієнтів із мінімальними сумарними витратами на перевезення. Час доставки немає значення. Передбачається, що всі операції по доставці товару можуть бути здійснені протягом деякого періоду часу, що влаштовує всіх споживачів.

За спосіб розвезення будемо вважати будь-яку припустиму комбінацію виконання замовлень. Він являє собою m–вимірний вектор-стовпець, i-й компонент якого дорівнює одиниці, якщо i–е замовлення цим способом задовольняється, і нулю – у протилежному випадку. Для будь-якої реальної задачі при невеликих значеннях m, k і r можна виписати всі такі способи розвезення. Число n цих способів буде залежати не тільки від перелічених параметрів, але й від кількості складів на кожному маршруті, об'єму замовлень і т. і. Кожному j–му способові розвезення відповідають грошові витрати c_j, $j=\overline{1,n}$.

Нехай при даних конкретних умовах задачі сформована матриця $A=\left[a_{ij}\right]$ всяких способів розвезення, що складається з нулів і одиниць. Стовпці цієї матриці являють собою описані вище способи розвезення, тобто елемент матриці $a_{ij} = 1$, якщо j–й спосіб задовольняє i–е замовлення, і $a_{ij} = 0$ – у протилежному випадку. Тепер задача полягає у виборі найбільш економічної комбінації цих способів.

Математична постановка задача

Уведемо змінні

$$x_j = \begin{cases} 1, & \text{якщо } j\text{ - й спосіб перевезення реалізується,} \\ 0, & \text{в протилежному випадку.} \end{cases} \qquad (1.39)$$

Тоді математична модель задачі набуває вигляду:

$$y = \sum_{j=1}^{n} c_j x_j \to \min_{x_j \in \Omega}, \qquad (1.40)$$

$$\Omega: \quad \sum_{j=1}^{n} a_{ij} x_j = 1, \quad i = \overline{1,m}, \qquad (1.41)$$

$$x_j \in \{0,1\}, \quad j = \overline{1,n}, \qquad (1.42)$$

$$a_{ij} \in \{0,1\}, \quad i = \overline{1,m}, \quad j = \overline{1,j}. \qquad (1.43)$$

Тут цільова функція (1.40) відповідає сумарним витратам на перевезення товару за всіма замовленнями. Умова (1.41) означає, що всі замовлення повинні бути задоволені тільки один раз. Вирази (1.42) і (1.43) визначають двійковий характер змінних x_j, $j = \overline{1,n}$, і елементів матриці **A.**

Приклад задачі про розвезення вантажу

Нехай у рамках умов задачі про розвезення вантажу відома матриця

$$\mathbf{A} = \left[a_{ij}\right] = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

усіх можливих способів розвезення вантажу з центральної бази в п'ять магазинів, а також витрати, що зв'язані з реалізацією кожного способу, а саме 11, 14, 9, 12, 13, 7,10, 8, 13 *ум. од*. Скласти графік розвезення, що забезпечує мінімальні сумарні витрати.

Математична модель задачі про розвезення вантажу в умовах прикладу

Математична модель задачі при використанні позначень, прийнятих для загальної моделі (1.39) – (1.43), буде мати вигляд:

$$y = 11x_1 + 14x_2 + 9x_3 + 12x_4 + 13x_5 +$$

$$+ 7x_6 + 10x_7 + 8x_8 + 13x_9 \to \min_{x_j \in \mathbf{\Omega}},$$

$$\mathbf{\Omega}: \quad f_1 = x_1 + x_3 + x_5 + x_6 + x_7 = 1,$$

$$f_2 = x_2 + x_5 + x_8 + x_9 = 1,$$

$$f_3 = x_2 + x_3 + x_4 + x_6 + x_8 = 1,$$

$$f_4 = x_1 + x_6 + x_8 + x_9 = 1,$$

$$f_5 = x_2 + x_3 + x_4 + x_5 + x_7 = 1,$$

$$x_j \in \{0,1\}, \quad j = \overline{1,9}.$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для транспортної задачі про розвезення вантажу в умовах прикладу можуть бути такими:

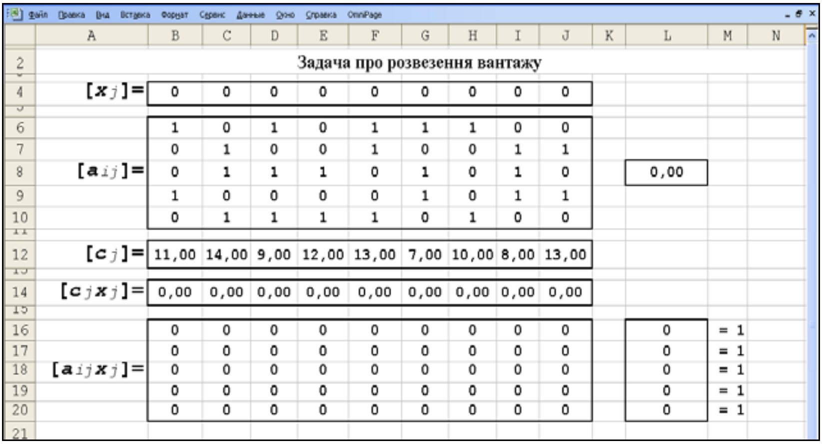
- клітинки B4:J4 – для шуканих змінних задачі x_j, $j=\overline{1,9}$;
- клітинки B6:J10 – для елементів a_{ij}, $i=\overline{1,5}$, $j=\overline{1,9}$, матриці **A;**
- клітинки B12:J12 – для задання розмірів витрат c_j на реалізацію *j*–го способу, $j=\overline{1,9}$;
- клітинки B14:J14 – для проміжних результатів із завантаженими формулами для обчислення добутку $c_j x_j$, $j=\overline{1,9}$;
- клітинки B16:J20 – для проміжних результатів із завантаженими формулами для обчислення добутку $a_{ij}x_j$, $i=\overline{1,5}$, $j=\overline{1,9}$;
- клітинки L16:L20 – для обчислення функцій f_i, $i=\overline{1,5}$, із відповідними завантаженими формулами;
- клітинка L8 – для значення цільової функції *y* із завантаженою формулою для обчислення суми $\sum_{j=1}^{9} c_j x_j$.

Всі клітинки, що призначені для цілочислових даних, повинні мати *числовий* формат із числом десяткових знаків, рівним 0. Це масиви клітинок: B4:J4, B6:J10, B16:J20 і L16:L20. Інші клітинки (L8, B12:J12 і B14:J14) повинні мати числовий формат із числом десяткових знаків, рівним 2.

Після завантаження в електронну таблицю всіх необхідних констант і формул для обчислення проміжних і кінцевого результатів треба виконати потрібні установки даних у діалоговому вікні команди *Сервіс/Пошук_рішення*. В умовах прикладу установки визначаються в такий спосіб:

- цільового клітинка – **$L$8**;
- тип екстремуму – **мінімум**;
- клітинки зі змінними: **$B$4:$J$4**;
- обмеження: **$B$4:$J$4 = двійкові**;
  $L$16:$L$20 = 1.

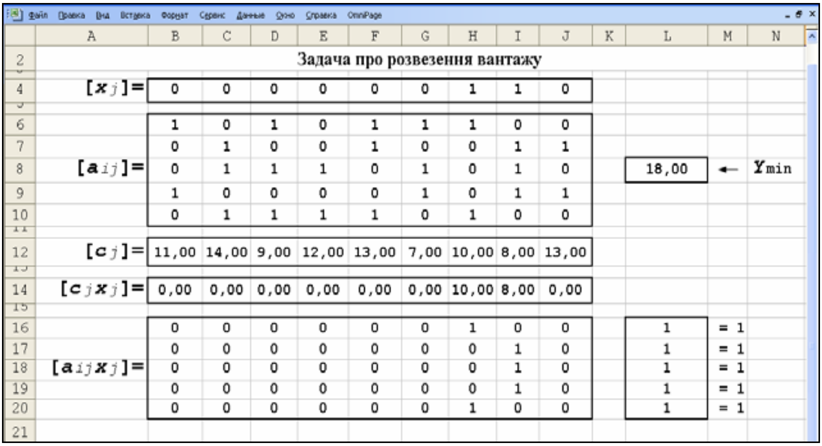
На рис.1.16 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
2						Задача про розвезення вантажу								
4	$[x_j]=$	0	0	0	0	0	0	0	0	0				
6		1	0	1	0	1	1	1	0	0				
7		0	1	0	0	1	0	0	1	1				
8	$[a_{ij}]=$	0	1	1	1	0	1	0	1	0		0,00		
9		1	0	0	0	0	1	0	1	1				
10		0	1	1	1	1	0	1	0	0				
12	$[c_j]=$	11,00	14,00	9,00	12,00	13,00	7,00	10,00	8,00	13,00				
14	$[c_j x_j]=$	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00				
16		0	0	0	0	0	0	0	0	0		0	= 1	
17		0	0	0	0	0	0	0	0	0		0	= 1	
18	$[a_{ij} x_j]=$	0	0	0	0	0	0	0	0	0		0	= 1	
19		0	0	0	0	0	0	0	0	0		0	= 1	
20		0	0	0	0	0	0	0	0	0		0	= 1	
21														

Рис. 1.16 – Екран з вихідними установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.17.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
2		Задача про розвезення вантажу												
4	[x_j]=	0	0	0	0	0	0	1	1	0				
6		1	0	1	0	1	1	1	0	0				
7		0	1	0	0	1	0	0	1	1				
8	[a_{ij}]=	0	1	1	1	0	1	0	1	0		18,00	←	Y_{min}
9		1	0	0	0	0	1	0	1	1				
10		0	1	1	1	1	0	1	0	0				
12	[c_j]=	11,00	14,00	9,00	12,00	13,00	7,00	10,00	8,00	13,00				
14	[$c_j x_j$]=	0,00	0,00	0,00	0,00	0,00	0,00	10,00	8,00	0,00				
16		0	0	0	0	0	0	1	0	0		1	= 1	
17		0	0	0	0	0	0	0	1	0		1	= 1	
18	[$a_{ij} x_j$]=	0	0	0	0	0	0	0	1	0		1	= 1	
19		0	0	0	0	0	0	0	1	0		1	= 1	
20		0	0	0	0	0	0	1	0	0		1	= 1	
21														

Рис. 1.17 – Екран з проміжними і кінцевими результатами

Оптимальним рішенням задачі є вектор-рядок $X^{*T} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0]$, що забезпечує мінімальне значення цільової функції $y^* = 18\,од.$, тобто увесь товар центрального складу треба розвести в магазини за допомогою 7-го та 8-го способів.

1.2.7. Задача про вибір маршруту

Змістовна постановка задачі

Транспортна мережа налічує (n + 1) пункт. Відомі відстані між пунктами c_{ij}, $i, j = \overline{0, n}$. Виїжджаючи з початкового пункту (йому приписується номер 0), комівояжер повинний побувати у всіх інших пунктах тільки один раз і повернутися в пункт 0. В якому порядку треба об'їжджати пункти, щоб пройдена сумарна відстань була мінімальною?

Примітка. Ця задача добре відома як задача про комівояжера.

Математична постановка задачі

Задачу комівояжера можна сформулювати як задачу цілочислового лінійного програмування. Впровадимо двійкові змінні x_{ij}, $i=\overline{0,n}$, $j=\overline{0,n}$; $i \neq j$, що мають наступний зміст:

$$x_{ij} = \begin{cases} 1, & \text{якщо комівояжер після пункту } i \text{ потрапляє в пункт } j, \\ 0 & \text{в протилежному випадку.} \end{cases}$$

Впровадимо також додаткові змінні $u_i, u_j \quad (i, j=\overline{1,n})$, що приймають довільні дійсні значення (не применшуючи спільності, їх можна вважати цілими). Дані змінні дозволяють сформувати умову зв'язності маршруту комівояжера: виключити розпадання маршруту на підцикли. Тоді математична модель задачі набуває вигляду:

$$\sum_{i=0}^{n} \sum_{\substack{j=0 \\ j \neq i}}^{n} c_{ij} x_{ij} \to \min_{x_{ij} \in \Omega}, \tag{1.44}$$

$$\Omega: \quad \sum_{\substack{j=0 \\ j \neq i}}^{n} x_{ij} = 1, \quad i=\overline{0,n}, \tag{1.45}$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} x_{ij} = 1, \quad j=\overline{0,n}, \tag{1.46}$$

$$u_i - u_j + n x_{ij} \le n-1, \;\; i, j=\overline{1,n}, \tag{1.47}$$

$$u_i, u_j = \text{int}, \;\; j, i=\overline{1,n}. \tag{1.48}$$

$$x_{ij} + x_{ji} \le 1, \;\; i, j=\overline{0,n}, \;\; i \neq j. \tag{1.49}$$

Тут (1.44) визначає цільову функцію як сумарну довжину маршруту комівояжера. Умова (1.45) говорить про те, що комівояжер в'їжджає в кожний пункт рівно один раз. Умова (1.46) говорить про те, що комівояжер виїжджає з кожного пункту рівно один раз. Обмеження (1.47) вимагає, щоб будь-який маршрут комівояжера складався з одного циклу. Система рівностей (1.48) обмежує область припустимих значень додаткових змінних цілими числами (позитивними чи негативними). Останнє обмеження (1.49) виключає повернення комівояжера до пункту, в якому він вже побував.

Приклад задачі про комівояжера

Є п'ять пунктів 0, 1, 2, 3, 4 (рис. 1.18), і надані відстані між ними у вигляді матриці

$$\mathbf{C} = [c_{ij}] = \begin{bmatrix} 0 & 12 & 4 & 9 & 10 \\ 12 & 0 & 11 & 6 & 8 \\ 4 & 11 & 0 & 13 & 5 \\ 9 & 6 & 13 & 0 & 7 \\ 10 & 8 & 5 & 7 & 0 \end{bmatrix}.$$

Виїжджаючи з пункту 0, комівояжер повинний побувати у всіх інших пунктах рівно один разі та повернутися до пункту 0. У якому порядку треба об'їжджати пункти, щоб пройдена сумарна відстань була мінімальною?

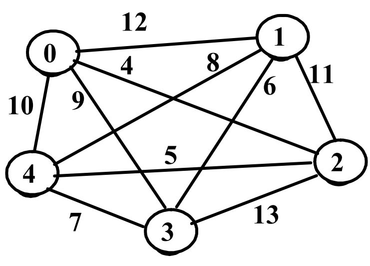


Рис. 1.18 – Умовна схема пунктів із указівкою відстаней між ними

Математична модель задачі про комівояжера в умовах прикладу

Математична модель задачі при використанні позначень, прийнятих для загальної моделі (1.44) – (1.49), буде мати вигляд:

$$y = 12x_{01} + 4x_{02} + 9x_{03} + 10x_{04} + 12x_{10} + 11x_{12} + 6x_{13} +$$
$$+ 8x_{14} + 4x_{20} + 11x_{21} + 13x_{23} + 5x_{24} + 9x_{30} + 6x_{31} +$$
$$+ 13x_{32} + 7x_{34} + 10x_{40} + 8x_{41} + 5x_{42} + 7x_{43} \to \min_{x_{ij} \in \boldsymbol{\Omega}} ,$$

$$\boldsymbol{\Omega}: \begin{aligned}
& f_0 = x_{01} + x_{02} + x_{03} + x_{04} = 1, \\
& f_1 = x_{10} + x_{12} + x_{13} + x_{14} = 1, \\
& f_2 = x_{20} + x_{21} + x_{23} + x_{24} = 1, \\
& f_3 = x_{30} + x_{31} + x_{32} + x_{34} = 1, \\
& f_4 = x_{40} + x_{41} + x_{42} + x_{43} = 1, \\
& f_5 = x_{10} + x_{20} + x_{30} + x_{40} = 1, \\
& f_6 = x_{01} + x_{21} + x_{31} + x_{41} = 1, \\
& f_7 = x_{02} + x_{12} + x_{32} + x_{42} = 1, \\
& f_8 = x_{03} + x_{13} + x_{23} + x_{43} = 1, \\
& f_9 = x_{04} + x_{14} + x_{24} + x_{34} = 1, \\
& u_i - u_j + 4x_{ij} \le 3, \quad i, j = \overline{1,4}, \ i \ne j \ , \\
& u_{_i}, u_i = \text{int}, \ i = \overline{1,4} \ . \\
& x_{ij} + x_{ji} \le 1, \ i, j = \overline{0,4}, \ i \ne j \ .
\end{aligned}$$

Цифрова модель і розв’язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для задачі про комівояжера в умовах прикладу може бути таким:

- клітинки C5:F5, B6, D6:F6, B7:C7, E7:F7, B8:D8, F8, B9:E9 – для шуканих значень змінних x_{ij}, $i, j = \overline{1,4}$;
- клітинки C11:F11 – для значень додаткових змінних u_j, $j = \overline{1,4}$;
- клітинки H5:H9 – для значень додаткових змінних u_i, $i = \overline{1,4}$;
- клітинки B14:F14 – для функцій f_i, $i = \overline{0,4}$, із відповідними завантаженими формулами;
- клітинки J5:J9 – для функцій f_{4+j}, $j = \overline{0,4}$, із відповідними завантаженими формулами;
- клітинки B16:F20 – для подання відстаней між пунктами c_{ij}, $i, j = \overline{0,4},\ i \neq j$;
- N6:P6, M7, O7:P7, M8:N8, P8, M9:O9 – для проміжних результатів із завантаженими формулами для обчислення виразу $u_i - u_j + 4x_{ij} \leq 3,\ i, j = \overline{1,4},\ i \neq j$,
- H11:K11, I12:K12, J13:K13, K14 – для проміжних результатів із завантаженими формулами для обчислення виразу $x_{ij} + x_{ji}$, $i, j = \overline{0,4},\ i \neq j$;
- H16:L20 – для проміжних результатів із завантаженими формулами для обчислення добутку $c_{ij}x_{ij}$, $i, j = \overline{0,4},\ i \neq j$;
- клітинка N18 – для цільової функції y із відповідною завантаженою формулою для її обчислення.

Всі клітинки електронної таблиці, що задіяні для розв’язання задачі, повинні мати *числовий* формат із числом десяткових знаків, рівним 0.

Установки даних в діалоговому вікні команди *Сервіс/Пошук_рішення* мають бути такими:

- для цільової клітинки – **$N$18**, що дорівнює **мінімальному значенню;**
- для клітинок із змінюваними змінними – **$C$5:$F$5**; **$B$6**; **$D$6:$F$6**; **$B$7:$C$7**; **$E$7:$F$7**; **$B$8:$D$8**; **$F$8**; **$B$9:$E$9**; **$C$11:$F$11**; **$H$5:$H$9**;
- для обмежень: **$B$14:$F$14 = 1**,
  $J$5:$J$9 = 1,
  $C$5:$F$5 = двійкове,
  $B$6 = двійкове,
  $B$7:$C$7 = двійкове ,
  $B$8:$D$8 = двійкове,
  $B$9:$E$9 = двійкове,
  $D$6:$F$6 = двійкове,
  $E$7:$F$7 = двійкове,
  $F$8 = двійкове,
  $B$11:$F$11 = ціле,
  $H$5:$H$9 = ціле,
  $N$6:$P$6 <= 3,
  $M$7 <= 3,
  $O$7:$P$7 <=3,
  $M$8:$N$8 <= 3,
  $P$8 <= 3,
  $M$9:$O$9 <= 3,
  $H$11:$K$11 <= 1,
  $I$12:$K$12 <= 1,
  $J$13:$K$13 <= 1,
  K$14 <= 1.

На рис.1.19 показаний вигляд екрана, що передує виконанню команди *Сервіс/Пошук_рішення*.

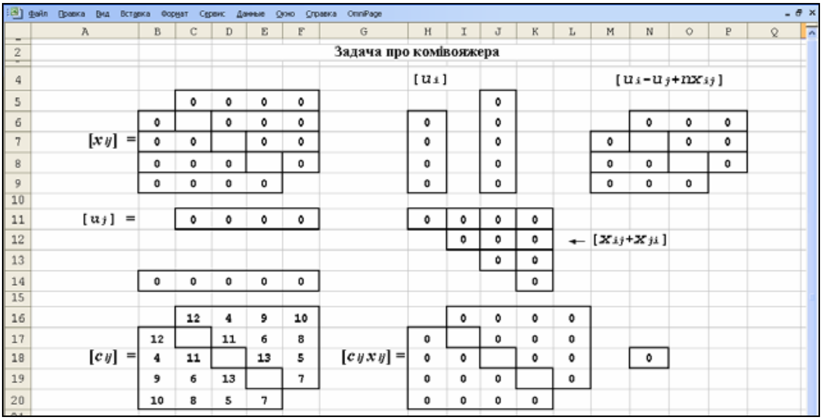


Рис. 1.19 – Екран з початковими установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.20.

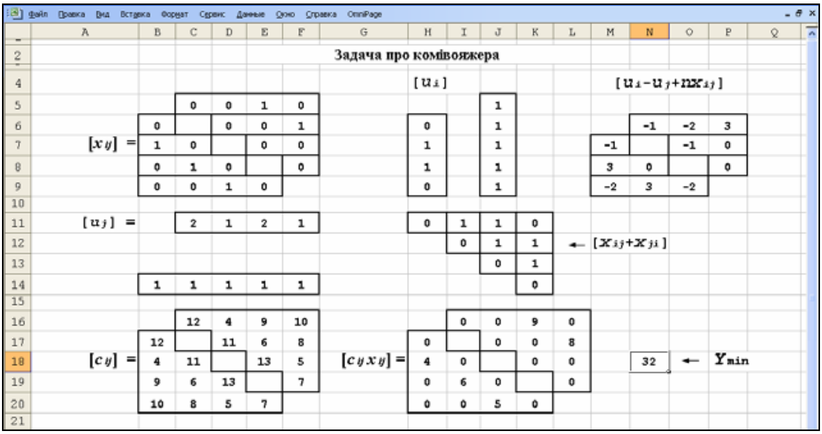


Рис. 1.20 – Екран з проміжними і кінцевими результатами

Рішення задачі $\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ визначає покриття, що відповідає графу на рис. 1.21. При цьому $y_{\min} = 32$.

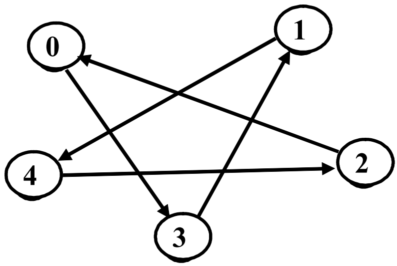


Рис. 1.21 – Умовна схема пунктів із вказівкою оптимального маршруту комівояжера

Примітка. У математичну модель задачі про комівояжера, коли її розглядають як задачу цілочислового лінійного програмування [14], не входить обмеження (1.49). Введення цього обмеження обумовлено тим, що задача вирішується в інформаційному середовищі *Microsoft Excel*. В загальному випадку без додання в математичну модель обмеження (1.49) задача про комівояжера в середовищі *Microsoft Excel* не розв'язується.

1.3. Задачі організації перевезень типу «багато-до-одного»

Задачі типу «багато-до-одного» в практиці вантажних перевезень виникають на транспортних і виробничих підприємствах, коли вантаж треба з декількох місць (підприємств, складів і т.

ін.) перевезти до одного замовника. Задачі цього типу складаються з таких класів:

- про планування випуску продукції;
- розподіл транспортних засобів за витратами часу;
- розподіл транспортних засобів за обсягом перевезень;
- розподіл транспортних засобів за витратами коштів;
- про розподіл транспортних засобів із фіксованими доплатами;
- про формування парку машин та їх розподіл;
- звезення вантажу;
- вибір маршруту.

Математичні моделі задач цього типу дуже схожі с математичними моделями задач типу «один-до-багатьох». За умовами задач обох типів перевезення вантажів відбувається в тих самих транспортних мережах. Відмінність спостерігаються тільки в напрямку руху. Якщо в задачах типу «один-до-багатьох» рух вантажу здійснюється в напрямку від кількох відправників (складів з сировиною, комплектуючими і таке інше) до одного замовника (центрального складу, підприємства), то в задачах типу «багато-до-одного» рух здійснюється ніби то в зворотному напрямку – від одного постачальника (виробника готової продукції, добувного підприємства, центрального складу) до декількох замовників (торговельних баз, складів, споживачів сировини). Якщо в одних задачах вантаж з одного місця розвозиться різними маршрутами між багатьма одержувачами, то в інших навпаки – звозиться різними маршрутами до одного місця. Отже, якщо умовно відправників вантажу вважати одержувачами, а замовника вантажу – відправником, то задача типу «багато-до-одного», перетворюється на задачу типу «один-до-багатьох». На підставі цього висновку розглядати зараз докладно всі класи задач типу «багато-до-одного» немає сенсу. Тому обмежимо подальший розгляд тільки задачею про планування випуску продукції, що не зустрічалася серед задач зі схемою перевезень «один-до-багатьох». Для інших класів наведемо тільки змістовну постанову.

1.3.1. Задача про планування випуску продукції

Змістовна постанова задачі

Для виготовлення n видів виробів на підприємстві використовується m видів сировини. Витрати *i*-го виду сировини на виготовлення *j*-го виду виробу в вагових одиницях становлять $a_{ij}, i=\overline{1,m}, \ j=\overline{1,n}$. Запаси *i*-го виду сировини зосереджені на *i*-му складі в розмірі b_i, $i=\overline{1,m}$, вагових одиниць. Витрати на перевезення однієї вагової одиниці сировини *i*-го виду зі складу на підприємство становлять c_i *ум. од.* Скласти оптимальний план виробництва, якщо прибуток підприємства від реалізації одного виробу *j*-го виду без врахування витрат на перевезення сировини становить d_j *ум. од.*, $j=\overline{1,n}$.

Математична модель задачі

Нехай x_j – кількість виробів *j*-го виду, $j=\overline{1,n}$, тоді задача про планування випуску продукції зводиться до задачі цілочислового лінійного програмування:

$$y=\sum_{j=1}^{n} d_j x_j - \sum_{i=1}^{m} c_i \sum_{j=1}^{n} a_{ij} x_j \to \max_{x_j \in \Omega}, \qquad (1.50)$$

$$\Omega: \quad \sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i=\overline{1,m}, \qquad (1.51)$$

$$x_j \ge 0; \quad x_j = \text{int}, \quad j=\overline{1,n}. \qquad (1.52)$$

Тут цільова функція (1.50) визначає розмір прибутку підприємства за відрахуванням витрат на перевезення сировини. Система нерівностей (1.51) обмежує витрати сировини її складською наявністю. Системи обмежень (1.52) визначають простір рішень.

Приклад задачі про планування випуску продукції

Для виготовлення п'яти видів виробів A_1, A_2, A_3, A_4, A_5 на підприємстві використовується чотири види сировини B_1, B_2, B_3, B_4. Витрати кожного виду сировини на виготовлення того або іншого виробу в вагових одиницях наведено в табл. 1.1.

Таблиця 1.1 – Дані до задачі про планування випуску продукції

Вид сировини	Витрати сировини на один виріб					Запаси	Витрати на перевезення
	A_1	A_2	A_3	A_4	A_5		
B_1	3	2	2	4	3	180	0,25
B_2	4	3	3	3	3	250	0,15
B_3	2	3	2	2	3	300	0,20
B_4	5	1	3	2	4	220	0,12
Питомий прибуток	8	6	6	7	7		

У таблиці також наведені запаси кожного типу сировини (в тих самих вагових одиницях), витрати (в тисячах умовних грошових одиницях) на перевезення однієї вагової одиниці сировини та питомий прибуток від реалізації одиниці продукції без урахування витрат на перевезення (теж в тисячах умовних грошових одиницях). Скласти оптимальний план виробництва.

Математична модель задачі про планування випуску продукції в умовах прикладу

Математична модель задачі при використанні позначень, що прийняті для загальної моделі цієї задачі (1.50) – (1.52), буде мати вигляд:

$$\begin{aligned} y = {} & 8x_1 + 6x_2 + 6x_3 + 7x_4 + 7x_5 - \\ & - 0{,}25\left(3x_1 + 2x_2 + 2x_3 + 4x_4 + 3x_5\right) - \\ & - 0{,}15\left(4x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5\right) - \\ & - 0{,}20\left(2x_1 + 3x_2 + 2x_3 + 2x_4 + 3x_5\right) - \\ & - 0{,}12\left(5x_1 + x_2 + 3x_3 + 2x_4 + 4x_5\right) \to \max_{x_j \in \Omega}, \end{aligned}$$

$$\Omega: \quad \begin{aligned} & f_1 = 3x_1 + 2x_2 + 2x_3 + 4x_4 + 3x_5 \le 180, \\ & f_2 = 4x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 \le 250, \\ & f_3 = 2x_1 + 3x_2 + 2x_3 + 2x_4 + 3x_5 \le 300, \\ & f_4 = 5x_1 + x_2 + 3x_3 + 2x_4 + 4x_5 \le 220, \\ & x_j \ge 0, \quad x_j = \text{int}, \quad j = \overline{1,5}. \end{aligned}$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для задачі про планування випуску продукції в умовах прикладу може бути таким:

- клітинки C3:G3 – для шуканих змінних задачі x_j, $j = \overline{1,5}$;
- клітинки C5:G8 – для задання витрат сировини a_{ij}, $i = \overline{1,4}$, $j = \overline{1,5}$;
- клітинки C10:G13 – для проміжних результатів із завантаженими формулами для обчислення добутку $a_{ij}x_j$, $i = \overline{1,4}$, $j = \overline{1,5}$;
- клітинки C15:G15 – для задання питомого прибутку d_j, $j = \overline{1,5}$;

- клітинки C17:G17 – для проміжних результатів із завантаженими формулами для обчислення добутку $d_j x_j$, $j = \overline{1,5}$;
- клітинки I10:I13 – для проміжних результатів із завантаженими формулами для обчислення суми $\sum_{j=1}^{5} a_{ij} x_{j,}$ $i = \overline{1,4}$;
- клітинки K10:K13 – для задання об’єму сировини на складі b_i, $i = \overline{1,4}$;
- клітинки M10:M13 – для задання витрат на перевезення сировини c_i, $i = \overline{1,4}$;
- клітинки O10:O13 – для проміжних результатів із завантаженими формулами для обчислення суми $c_i \sum_{j=1}^{5} a_{ij} x_{j,}$ $i = \overline{1,4}$;
- клітинка I3 – для проміжного результату із завантаженою формулою для обчислення суми $\sum_{j=1}^{5} d_j x_j$;
- клітинка K3 – для проміжного результату із завантаженою формулою для обчислення подвійної суми $\sum_{i=1}^{4} c_i \sum_{j=1}^{5} a_{ij} x_j$;
- клітинка M3 – для значення цільової функції y із завантаженою формулою для обчислення виразу $\sum_{j=1}^{5} d_j x_j - \sum_{i=1}^{4} c_i \sum_{j=1}^{5} a_{ij} x_j$.

Всі клітинки, що призначені для цифрової моделі, повинні мати *числовий* формат із числом десяткових знаків, рівним 2. Виключення складають клітинки з цілочисловими даними. Це –

клітинки C3:G3 для шуканих змінних задачі. Вони повинні мати числовий формат із числом десяткових знаків, рівним 0.

Установки даних в діалоговому вікні команди *Сервіс/Пошук_рішення*:

цільова клітинка – **$M$3**;

тип екстремуму – **максимум**;

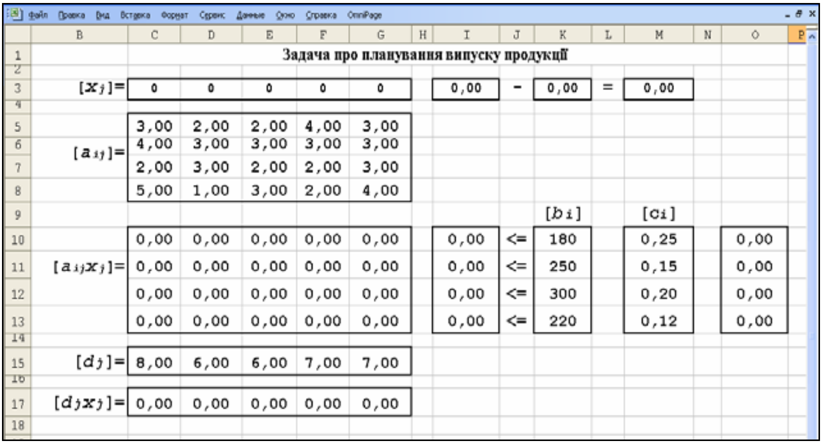
клітинки зі змінними – **$C$3:$G$3**;

обмеження: **$C$3:$G$3 = ціле**;

$C$3:$G$3 >= 0;

$I$10:$I$13 <= $K$10:$K$13;

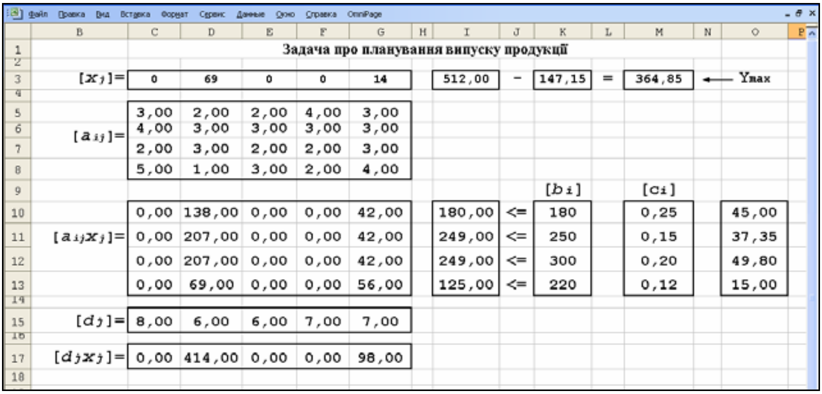
На рис.1.22 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.



	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		Задача про планування випуску продукції												
2														
3	[xj]=	0	0	0	0	0		0,00	-	0,00	=	0,00		
4														
5		3,00	2,00	2,00	4,00	3,00								
6	[aij]=	4,00	3,00	3,00	3,00	3,00								
7		2,00	3,00	2,00	2,00	3,00								
8		5,00	1,00	3,00	2,00	4,00								
9										[bi]		[ci]		
10		0,00	0,00	0,00	0,00	0,00		0,00	<=	180		0,25		0,00
11	[aijxj]=	0,00	0,00	0,00	0,00	0,00		0,00	<=	250		0,15		0,00
12		0,00	0,00	0,00	0,00	0,00		0,00	<=	300		0,20		0,00
13		0,00	0,00	0,00	0,00	0,00		0,00	<=	220		0,12		0,00
14														
15	[dj]=	8,00	6,00	6,00	7,00	7,00								
16														
17	[djxj]=	0,00	0,00	0,00	0,00	0,00								
18														

Рис. 1.22 – Екран з початковими установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.23.



	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		Задача про планування випуску продукції												
2														
3	[xj]=	0	69	0	0	14		512,00	-	147,15	=	364,85	←	Ymax
4														
5		3,00	2,00	2,00	4,00	3,00								
6	[aij]=	4,00	3,00	3,00	3,00	3,00								
7		2,00	3,00	2,00	2,00	3,00								
8		5,00	1,00	3,00	2,00	4,00								
9										[bi]		[ci]		
10		0,00	138,00	0,00	0,00	42,00		180,00	<=	180		0,25		45,00
11	[aijxj]=	0,00	207,00	0,00	0,00	42,00		249,00	<=	250		0,15		37,35
12		0,00	207,00	0,00	0,00	42,00		249,00	<=	300		0,20		49,80
13		0,00	69,00	0,00	0,00	56,00		125,00	<=	220		0,12		15,00
14														
15	[dj]=	8,00	6,00	6,00	7,00	7,00								
16														
17	[djxj]=	0,00	414,00	0,00	0,00	98,00								
18														

Рис. 1.23 – Екран з проміжними і кінцевими результатами

Оптимальним розв'язком задачі є вектор-рядок $\mathbf{X}^{*T} = \begin{bmatrix} 0 & 69 & 0 & 0 & 14 \end{bmatrix}$. Компоненти вектору $\mathbf{X}^*$ свідчать про те, що за умовами задачі оптимальний план виробництва припускає випуск 69 виробів 2-го типу та 14 виробів 5-го типу. При цьому буде вичерпана повністю сировина 1-го типу. Інша сировина буде мати незначні залишки. При такому плані підприємство отримає максимальний прибуток у розмірі $y^* = 364{,}85$ *тис. ум. од.*

1.3.2. Змістовні постановки задач зі схемою перевезень типу «багато-до-одного»

Задача про розподіл транспортних засобів за витратами часу

Для перевезення піску з n піщаних кар'єрів (за n транспортними маршрутами) на один будівельний майданчик використовуються транспортні засоби m типів. Продуктивність i-ї транспортної одиниці на j–му маршруті становить t_{ij} тонн за добу, $i = \overline{1,m},\ j = \overline{1,n}$. Резерв корисного часу i-го транспортного засобу обмежений величиною a_i діб, $i=\overline{1,m}$. Мінімальний об'єм вантажу, що перевозиться на j–му маршруті становить b_j тонн, $j=\overline{1,n}$. Визначити, які саме транспортні засоби, на яких маршрутах і на протязі якого терміну (скількох діб) потрібно використовувати, щоб забезпечити перевезення всього вантажу при мінімальних витратах загального резерву корисного часу. Вважається, що кожний транспортний засіб на протязі однієї доби може обслуговувати тільки один маршрут, тобто кількість часу вимірюється цілим числом діб.

Якщо через x_{ij} позначити кількість діб, на протязі яких транспортний засіб i-го типу використовується на j-му маршруті, то математична модель задачі повністю співпадає з моделлю задачі зі схемою перевезень «один-до-багатьох» про розподіл транспортних засобів за витратами часу (1.12) – (1.16).

Задача про розподіл транспортних засобів за обсягом перевезень

Для перевезення піску з n піщаних кар'єрів (за n транспортними маршрутами) на один будівельний майданчик використовуються транспортні засоби m типів. Продуктивність i-ї транспортної одиниці на j–му маршруті становить t_{ij} тонн за добу, $i=\overline{1,m},\ j=\overline{1,n}$. Резерв корисного часу i-го транспортного засобу обмежений величиною a_i *діб*, $i=\overline{1,m}$. Мінімальний об'єм вантажу, що перевозиться на j–му маршруті становить b_j тонн, $j=\overline{1,n}$. Визначити, які саме транспортні засоби, на яких маршрутах і на протязі якого терміну (скількох діб) потрібно використовувати, щоб забезпечити перевезення максимально можливого об'єму вантажу. Вважається, що кожний транспортний засіб на протязі однієї доби може обслуговувати тільки один маршрут.

Якщо через x_{ij} позначити кількість діб, на протязі яких транспортний засіб i-го типу використовується на j-му маршруті, то математична модель задачі повністю співпадає з моделлю задачі зі схемою перевезень «один-до-багатьох» про розподіл транспортних засобів за обсягом перевезень (1.17) – (1.21).

Задача про розподіл транспортних засобів за грошовими витратами

Нехай є n транспортних ліній, за якими щоденно звозять сировину до житлово-будівельного комбінату. На j-й лінії необхідно виконати b_j рейсів, $j = \overline{1,n}$. У наявності є транспортні одиниці m типів. Резерви корисного часу транспортних одиниць типу i становлять a_i, $i = \overline{1,m}$. На виконання транспортною одиницею типу i рейсу на лінії j потрібен час t_{ij}, а грошові витрати на один рейс становлять c_{ij}. Потрібно визначити найбільш економічне розміщення транспортних одиниць за всіма лініями, тобто розміщення з мінімальними грошовими витратами на виконання всіх рейсів.

Якщо через x_{ij} позначити кількість рейсів, що транспортна одиниця i повинна виконати на лінії j, то математична модель задачі повністю співпадає з моделлю задачі зі схемою перевезень «один-до-багатьох» про розподіл транспортних засобів за грошовими витратами (1.22) – (1.26).

Задача про розподіл транспортних засобів із фіксованими доплатами

Нехай є n транспортних ліній, за якими щоденно звозять сировину до житлово-будівельного комбінату. На j–й лінії необхідно виконати b_j рейсів, $j = \overline{1,n}$. У наявності є транспортні одиниці m типів. Резерви корисного часу транспортних одиниць типу *i* становлять $a_i, i=\overline{1,m}$. На виконання транспортною одиницею типу *i* рейсу на лінії *j* потрібен час t_{ij}, а грошові витрати на один рейс становлять c_{ij}. Потрібно визначити найбільш економічне розміщення транспортних одиниць по лініях, тобто мінімальні грошові витрати на виконання всіх рейсів. Крім того, на відміну від попередньої задачі про розподіл транспортних засобів за грошовими витратами, випуск транспортної одиниці типу *i* на лінію *j* пов'язаний із підготовчими роботами, що вимагають витрат часу в розмірі t_{ij}^{+}. Цей час не залежить від числа рейсів, що повинен виконати транспортний засіб. Грошові витрати на проведення цих підготовчих робіт становлять c_{ij}^{+}. Потрібно визначити найбільш економічне розміщення транспортних одиниць за всіма лініями.

Якщо через x_{ij} позначити кількість рейсів, що транспортна одиниця *i* повинна виконати на лінії *j*, то математична модель задачі повністю співпадає с моделлю задачі зі схемою перевезень «один-до-багатьох» про розподіл транспортних засобів із фіксованими доплатами (1.27) – (1.33).

Задача про формування парку машин та їх розподіл

Маємо *n* типів транспортних засобів і *m* видів сировини, що треба перевезти відповідно з *m* складів до виробничого комбінату в об'ємі b_i вагових одиниць, $i=\overline{1,m}$. Задана вантажомісткість *j*-го транспортного засобу відносно на *i*-го виду сировини $a_{ij}, i=\overline{1,m}, j=\overline{1,n}$, а також грошові витрати c_{ij} на перевезення одиниці сировини *i*-го виду транспортним засобом *j*-го типу. Відомо, що розмір оренди одного транспортного засобу *j*-го типу становить d_j *ум.од.*, $j=\overline{1,n}$. Треба визначити такий машинний парк для перевезення всіх видів сировини до виробничого комбінату із подальшим розподілом машин між роботами, щоб виконати всі перевезення з мінімальними грошовими витратами.

Якщо позначити через x_j кількість транспортних засобів *j*–го типа, $j = \overline{1,n}$, а через y_{ij} – кількість транспортних засобів *j*–го типу, що залучаються до перевезення *i*–го виду сировини, то математична модель задачі повністю співпадає з моделлю задачі зі схемою перевезень «один-до-багатьох» про формування парку машин та їх розподіл (1.34) – (1.38).

Задача про звезення вантажу

До деякої центральної бази звозиться товар (його можна вважати однорідним) від m виробників. Звезення товару на центральну базу здійснюється однією вантажівкою, причому від кожного виробника товар вивозиться цілком за один прийом – вантажопідйомність вантажівки для цього достатня. Вантажівка може одночасно взяти вантаж, не більш ніж від *k* виробників. Вантажівка може об'їжджати виробників за *r* маршрутами. До одного й того самого виробника можна потрапити різними маршрутами. Відомі грошові витрати c_j, $j = \overline{1,n}$, на виконання перевезення товару за тим або іншим маршрутом. Потрібно вибрати маршрути звезення всього товару на центральну базу із мінімальними сумарними витратами. Час доставки немає значення. Передбачається, що всі операції по звезенню товару можуть бути здійснені протягом деякого періоду часу, що влаштовує всіх споживачів.

Якщо ввести змінні x_j, $j = \overline{1,n}$, які приймають значення 1, коли для звезення товару використовується *j*–й маршрут, і значення 0 – в протилежному випадку, то математична модель задачі повністю співпадає з моделлю задачі зі схемою перевезень «один-до-багатьох» про розвезення вантажу (1.40) – (1.43). При цьому параметр a_{ij} в (1.41) і (1.43) дорівнює 1, якщо *j*–й маршрут проходить через *i*–го виробника товару, і дорівнює 0 – в протилежному випадку.

Задача про вибір маршруту

Транспортна мережа налічує (n + 1) пункт. Відомі відстані між пунктами $c_{ij}, i, j = \overline{0,n}$. В одному пункті (йому приписується номер 0) знаходиться підприємство зі складання певного пристрою, в інших пунктах зосереджено окремі види комплектуючих виробів для цього пристрою. Виїжджаючи з пункту 0, вантажівка повинна побувати у всіх інших пунктах тільки один раз, завантажити комплектуючі вироби й повернутися в пункт 0. В якому порядку треба вантажівці об'їжджати пункти, щоб пройдена сумарна відстань була мінімальною?

Введемо двійкові змінні x_{ij}, $i = \overline{0,n}$, $j = \overline{0,n}$; $i \neq j$, що мають наступний зміст:

$$x_{ij} = \begin{cases} 1, & \text{якщо вантажівка після пункту } i \text{ потрапляє в пункт } j, \\ 0 & \text{в протилежному випадку.} \end{cases}$$

Тоді математична модель задачі повністю співпадає з моделлю задачі комівояжера (1.44) – (1.49), що була розглянута серед задач зі схемою перевезень «один-до-багатьох».

1.4. Задачі зі схемою перевезень «багато-до-багатьох»

Задачі типу «багато-до-багатьох» в практиці вантажних перевезень виникають на транспортних і виробничих підприємствах, коли вантаж треба з декількох місць (підприємств, складів і т. ін.) перевезти до декількох замовників. Саме цьому типу належить класична задача про організацію перевезення, що відома як транспортна задача.

Серед задач зі схемою перевезень «багато-до-багатьох» розглянемо такі:

- класична транспортна задача;
- відкрита транспортна задача з неперервною математичною моделлю;
- транспортна задача з дискретною математичною моделлю;
- перевезення вантажу в два етапи;
- перевезення вантажу декількох видів у два етапи;
- перевезення вантажу декількох видів у два етапи за замовленням споживачів;
- закриття підприємства;
- розиграш кубку.

1.4.1. Класична транспортна задача

Змістовна постановка задачі

Загальна постановка транспортної задачі організації перевезень із неперервною математичною моделлю полягає у визначенні оптимального плану перевезень деякого однорідного вантажу з m пунктів виробництва $A_1, A_2, \ldots, A_m$ у n пунктів споживання $B_1, B_2, \ldots, B_n$. Для кожного пункту відправлення задані об'єми виробництва $a_1, a_2, \ldots, a_m$. Для кожного пункту призначення задані об'єми споживання $b_1, b_2, \ldots, b_n$. Потрібно скласти такий план перевезень, що цілком забезпечив би всіх споживачів при мінімальних витратах на перевезення. При цьому в якості критерію оптимальності зазвичай береться або мінімальна вартість перевезень усього вантажу, або мінімальний час його доставки, або мінімальний сумарний пробіг вантажного транспорту.

Вважається, що загальна потреба у вантажі в пунктах призначення не перевищує сумарних запасів вантажу в пунктах відправлення. У противному випадку задовольнити потреби споживачів принципово неможливо.

Математична постановка задачі

Уведемо позначення:

- c_{ij} – витрати на перевезення одиниці вантажу з i–го пункту відправлення в j–й пункт призначення;
- a_i – запаси вантажу в i–му пункті відправлення;
- b_j – потреба у вантажі в j–му пункті призначення;
- x_{ij} – кількість одиниць вантажу, що перевозиться з i–го пункту відправлення в j–й пункт призначення.

Тоді відкрита математична модель транспортної задачі має вигляд:

$$y = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min_{x_{ij} \in \Omega \subset E^n}, \tag{1.53}$$

$$\Omega: \quad f_i = \sum_{j=1}^{n} x_{ij} = a_i, \quad i = \overline{1,m}, \tag{1.54}$$

$$f_{m+j} = \sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1,n}, \tag{1.55}$$

$$x_{ij} \geq 0; \quad i = \overline{1,m}; \quad j = \overline{1,n}. \tag{1.56}$$

Вираз (1.53) – це цільова функція, що визначає вартість перевезень усього вантажу. Саме екстремальне (мінімальне) значення цієї функції необхідно знайти в процесі розв’язання задачі. Крім того, треба визначити значення змінних x_{ij}, при яких цільова функція досягає свого мінімуму. Причому ці значення повинні належати неперервної області припустимих рішень Ω.

Вирази (1.54) – (1.56) визначають область припустимих рішень Ω. Так, система рівностей (1.54) вимагає, щоб сумарні об’єми перевезень співпадали з запасами вантажу в пунктах відправлення. Система рівностей (1.55) забезпечує потреби в вантажу в кожному пункті призначення. Система нерівностей (1.56) визначає припустимий простір змінних x_{ij}, вона відсікає від’ємну область простору змінних $\mathbf{E}^n$, у яку зміні не можуть потрапляти за своєю фізичною суттю.

Функції y, $f_1, f_2, \dots, f_{m+n}$ є неперервними лінійними функціями, заданими на невід’ємному октанті евклідова простору $\mathbf{E}^n$. Дані функції мають місце, коли вантаж є рідиною, сипкими речовинами, дрібними комплектуючими заготівлями або дрібною непартійною продукцією. Такий вантаж характеризується параметрами, що являють собою вагу, погонні метри, квадратні метри, об'єм і т.п., але не штуки, упаковки, партії тощо.

Якщо загальна потреба у вантажі в пунктах призначення дорівнює сумарному запасу вантажу в пунктах відправлення, тобто

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \quad , \tag{1.57}$$

то модель такої транспортної задачі вважається закритою або збалансованою, а сама задача – класичною.

Умова (1.57) у явному вигляді не є присутньою в обмеженнях транспортної задачі. Вона неявно враховується системами обмежень (1.54) та (1.55).

Приклад транспортної задачі

Нехай чотири підприємства для виробництва продукції використовують три види сировини. Потреби в сировині кожного з підприємств відповідно рівні 120,5; 49,5; 189,2; 110,8 вагових одиниць. Сировина зосереджена на трьох складах відповідно в кількостях 160, 140, 170 вагових одиниць. На кожне підприємство сировина може завозитися з будь-якого складу. Тарифи перевезень є відомими і задаються матрицею $\mathbf{C} = \begin{bmatrix} 7 & 8 & 1 & 2 \\ 4 & 5 & 9 & 8 \\ 9 & 2 & 3 & 6 \end{bmatrix}$ (*ум. од.*). Необхідно скласти такий план перевезень, за яким загальні витрати на доставку сировини на підприємства є мінімальними.

Математична модель транспортної задачі в умовах прикладу

Перед укладанням математичної моделі перевіряємо умову (1.52), щоб переконатися в тому, що математична модель задачі є закритою. Дійсно, загальна потреба у вантажі 470 *од.* (120,5 + 49,5 + 189,2 +110,8) дорівнює сумарним запасам 470 *од*. (160 + 140 + 170).

Математична модель задачі в умовах прикладу з використанням прийнятих позначень для задачі (1.53) – (1.56) має вигляд:

$$y = 7x_{11} + 8x_{12} + x_{13} + 2x_{14} + 4x_{21} + 5x_{22} + 9x_{23} + \\ + 8x_{24} + 9x_{31} + 2x_{32} + 3x_{33} + 6x_{34} \to \min_{x_{ij} \in \mathbf{\Omega}},$$

$$\mathbf{\Omega}: \quad f_1 = x_{11} + x_{12} + x_{13} + x_{14} = 160;$$

$$f_2 = x_{21} + x_{22} + x_{23} + x_{24} = 140;$$

$$f_3 = x_{31} + x_{32} + x_{33} + x_{34} = 170;$$

$$f_4 = x_{11} + x_{21} + x_{31} = 120{,}5;$$

$$f_5 = x_{12} + x_{22} + x_{32} = 49{,}5;$$

$$f_6 = x_{13} + x_{23} + x_{33} = 189{,}2;$$

$$f_7 = x_{14} + x_{24} + x_{34} = 110{,}8;$$

$$x_{ij} \ge 0; \quad i = \overline{1,3}; \quad j = \overline{1,4}.$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для транспортної задачі можуть бути такими:

- клітинки B5:E7 – для шуканих змінних задачі x_{ij}, $i = \overline{1,3}$; $j = \overline{1,4}$
- клітинки B9:E9 – для функцій f_{3+j}, $j = \overline{1,4}$, із завантаженими формулами для обчислення сум $x_{1j} + x_{2j} + x_{3j}$, $j = \overline{1,4}$;
- клітинки B11:E11 – для задання потреб у вантажу b_j, $j = \overline{1,4}$, у пунктах призначення;
- клітинки G5:G7 – для функцій f_i, $i = \overline{1,3}$, із завантаженими формулами для обчислення сум $x_{i1} + x_{i2} + x_{i3} + x_{i4}$, $i = \overline{1,3}$;
- клітинки I5:I7 – для задання запасів вантажу a_i, $i = \overline{1,3}$, у пунктах відправлення;
- клітинки B13:E15 – для елементів тарифної матриці **C**;

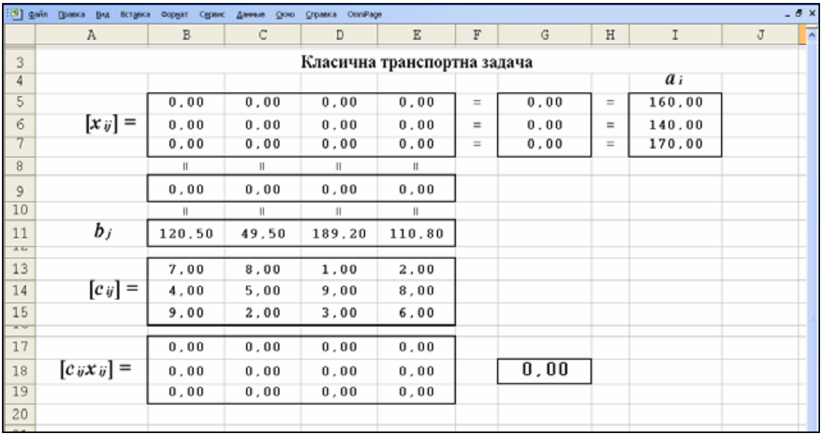
- клітинки B17:E19 – для проміжних результатів із завантаженими формулами для обчислення добутку $c_{ij}x_{ij}$, $i=\overline{1,3}$, $j=\overline{1,4}$;
- клітинка G18 – для значення цільової функції *y* із завантаженою формулою для обчислення подвійної суми $\sum_{i=1}^{3}\sum_{j=1}^{4}c_{ij}x_{ij}$.

Всі клітинки, що задіяні в цифровій моделі, повинні бути відформатовані. Для них треба вибрати *числовий* формат. Число десяткових знаків для числового формату установлюється, виходячи з характеру одиниць виміру. Вибір повинний забезпечувати достатню точність обчислення. Так, для цільовою клітинки, що призначена для грошової величини, установлюється формат із двома десятковими знаками. У цьому прикладі для всіх клітин умовно прийнятий такий же формат.

Після завантаження в клітинки всіх необхідних констант і формул для обчислення проміжних і кінцевого результатів треба виконати необхідні установки даних у діалоговому вікні команди *Сервіс/Пошук_рішення*. В умовах приклада установки визначаються в такий спосіб:

- цільова клітинка – **$G$18;**
- тип екстремуму – **мінімум;**
- клітинки для змінних задачі – **$B$5:$E$7;**
- обмеження задачі: **$B$5:$E$7 >= 0;**
  $B$9:$E$9 = $B$11:$E$11;
  $G$5:$G$7 = $I$5:$I$7.

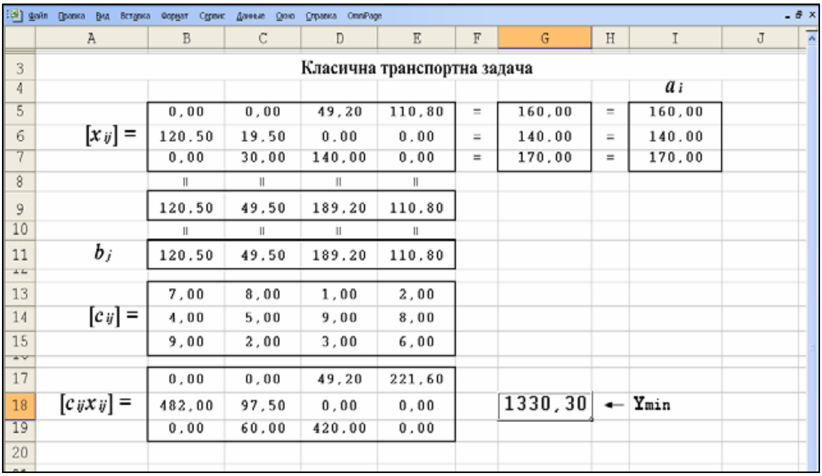
На рис.1.24 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.



	A	B	C	D	E	F	G	H	I
3				Класична транспортна задача					
4									a_i
5		0,00	0,00	0,00	0,00	=	0,00	=	160,00
6	$[x_{ij}]$ =	0,00	0,00	0,00	0,00	=	0,00	=	140,00
7		0,00	0,00	0,00	0,00	=	0,00	=	170,00
8		॥	॥	॥	॥				
9		0,00	0,00	0,00	0,00				
10		॥	॥	॥	॥				
11	b_j	120,50	49,50	189,20	110,80				
13		7,00	8,00	1,00	2,00				
14	$[c_{ij}]$ =	4,00	5,00	9,00	8,00				
15		9,00	2,00	3,00	6,00				
17		0,00	0,00	0,00	0,00				
18	$[c_{ij}x_{ij}]$ =	0,00	0,00	0,00	0,00		0,00		
19		0,00	0,00	0,00	0,00				

Рис. 1.24 – Екран із вихідними установками

Вигляд екрана після виконання команди *Сервіс/Пошук_р-ішення* показаний на рис. 1.25.



	A	B	C	D	E	F	G	H	I
3				Класична транспортна задача					
4									a_i
5		0,00	0,00	49,20	110,80	=	160,00	=	160,00
6	$[x_{ij}]$ =	120,50	19,50	0,00	0,00	=	140,00	=	140,00
7		0,00	30,00	140,00	0,00	=	170,00	=	170,00
8		॥	॥	॥	॥				
9		120,50	49,50	189,20	110,80				
10		॥	॥	॥	॥				
11	b_j	120,50	49,50	189,20	110,80				
13		7,00	8,00	1,00	2,00				
14	$[c_{ij}]$ =	4,00	5,00	9,00	8,00				
15		9,00	2,00	3,00	6,00				
17		0,00	0,00	49,20	221,60				
18	$[c_{ij}x_{ij}]$ =	482,00	97,50	0,00	0,00		1330,30	←	Y_{min}
19		0,00	60,00	420,00	0,00				

Рис. 1.25 – Екран з проміжними і шуканими результатами

Оптимальним рішенням задачі є матриця значень змінних $\mathbf{X}^* = \begin{bmatrix} 0{,}0 & 0{,}0 & 49{,}2 & 110{,}8 \\ 120{,}5 & 19{,}5 & 0{,}0 & 0{,}0 \\ 0{,}0 & 30{,}0 & 140{,}0 & 0{,}0 \end{bmatrix}$, що забезпечує мінімум цільової функції $y^* = 1330{,}3$ *ум. од.*

1.4.2. Транспортна задача із неперервною відкритою математичною моделлю

Змістовна і математична постановки задачі

Загальна постановка транспортної задачі із неперервною відкритою математичною моделлю майже цілком збігається з постановкою класичної транспортної задачі. Відмінність полягає лише в системі обмежень (1.49), яка зараз набуває вигляду

$$f_i = \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = \overline{1,m} \,, \tag{1.58}$$

Таким чином, в загальному вигляді математична модель неперервної відкритої транспортної задачі складається з цільової функції (1.53) і систем обмежень (1.55) , (1.56) і (1.58).

Система нерівностей (1.58) при інших рівних умовах спричиняє значне збільшення області припустимих значень (в порівнянні із закритою математичною моделлю транспортної задачі), тому цільова функція зазвичай досягає кращих значень. Наступний приклад надає можливість переконатися в цьому твердженні.

Приклад транспортної задачі із неперервною відкритою математичною моделлю

Нехай чотири підприємства для виробництва продукції використовують три види сировини. Потреби в сировині кожного з підприємств відповідно рівні 120,5; 49,5; 189,2; 110,8 вагових одиниць. Сировина зосереджена на трьох складах відповідно в кількостях 170, 150, 180 вагових одиниць. На кожне підприємство сировина може завозитися з будь-якого складу. Тарифи перевезень є відомими і задаються матрицею $\mathbf{C} = \begin{bmatrix} 7 & 8 & 1 & 2 \\ 4 & 5 & 9 & 8 \\ 9 & 2 & 3 & 6 \end{bmatrix}$ (*ум. од.*). Необхідно скласти такий план перевезень, за яким загальні витрати на доставку сировини на підприємства є мінімальними.

Умови наведеного прикладу відрізняється від умов попереднього прикладу лише об'ємами визначальних запасів сировини. Зараз загальний об'єм сировини на складах (170 + 150 + + 180 = 500 *од.*) перевищують потреби підприємств (120.5 + + 49,5 + 189,2 + 110,8 = 470 *од.*). Тобто математична модель цієї транспортної задачі є відкритою.

Математична модель відкритої транспортної задачі

Математична модель відкритої транспортної задачі в умовах прикладу зі збереженням позначень, що використовувалися для загальної моделі (1.49) – (1.52), набуває вигляду:

$$y = 7x_{11} + 8x_{12} + x_{13} + 2x_{14} + 4x_{21} + 5x_{22} + 9x_{23} + \\ + 8x_{24} + 9x_{31} + 2x_{32} + 3x_{33} + 6x_{34} \to \min_{x_{ij} \in \mathbf{\Omega}},$$

$$\mathbf{\Omega}: \quad f_1 = x_{11} + x_{12} + x_{13} + x_{14} \le 170;$$

$$f_2 = x_{21} + x_{22} + x_{23} + x_{24} \le 150\,;$$

$$f_3 = x_{31} + x_{32} + x_{33} + x_{34} \le 180\,;$$

$$f_4 = x_{11} + x_{21} + x_{31} = 120{,}5\,;$$

$$f_5 = x_{12} + x_{22} + x_{32} = 49{,}5\,;$$

$$f_6 = x_{13} + x_{23} + x_{33} = 189{,}2\,;$$

$$f_7 = x_{14} + x_{24} + x_{34} = 110{,}8\,;$$

$$x_{ij} \ge 0; \quad i = \overline{1,3}; \quad j = \overline{1,4}\,.$$

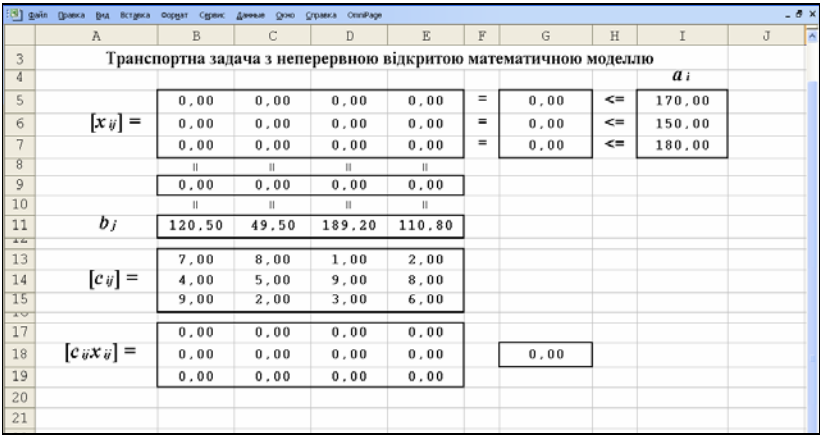
Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для транспортного задачі можуть бути такими, як і в попередньому прикладі. Однак установки в діалоговому вікні команди будуть дещо іншими:

- цільова клітинка – **$G$18;**
- тип екстремуму – **мінімум;**
- клітинки для змінних задачі – **$B$5:$E$7;**
- обмеження задачі: **$B$5:$E$7 >= 0;**
  $B$9:$E$9 = $B$11:$E$11;
  $G$5:$G$7 <= $I$5:$I$7.

Тут слід звернути увагу на останнє обмеження, що на відміну від аналогічного обмеження в попередньому прикладі, являє собою систему нерівностей.

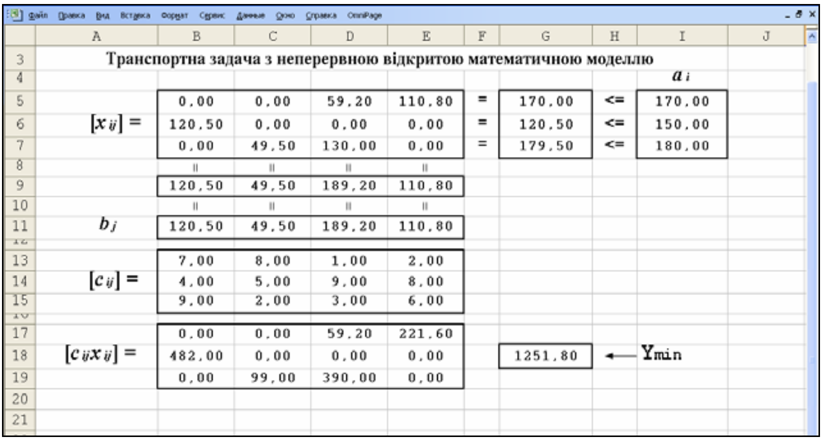
На рис.1.26 показаний вигляд екрана, що має передувати виконанню команди *Сервіс/Пошук_рішення*.



	A	B	C	D	E	F	G	H	I
3	Транспортна задача з неперервною відкритою математичною моделлю								
4									a_i
5		0,00	0,00	0,00	0,00	=	0,00	<=	170,00
6	$[x_{ij}]$ =	0,00	0,00	0,00	0,00	=	0,00	<=	150,00
7		0,00	0,00	0,00	0,00	=	0,00	<=	180,00
8		||	||	||	||				
9		0,00	0,00	0,00	0,00				
10		||	||	||	||				
11	b_j	120,50	49,50	189,20	110,80				
13		7,00	8,00	1,00	2,00				
14	$[c_{ij}]$ =	4,00	5,00	9,00	8,00				
15		9,00	2,00	3,00	6,00				
17		0,00	0,00	0,00	0,00				
18	$[c_{ij}x_{ij}]$ =	0,00	0,00	0,00	0,00		0,00		
19		0,00	0,00	0,00	0,00				

Рис. 1.26 – Екран із початковими установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.27.



	A	B	C	D	E	F	G	H	I
3	Транспортна задача з неперервною відкритою математичною моделлю								
4									a_i
5		0,00	0,00	59,20	110,80	=	170,00	<=	170,00
6	$[x_{ij}]$ =	120,50	0,00	0,00	0,00	=	120,50	<=	150,00
7		0,00	49,50	130,00	0,00	=	179,50	<=	180,00
8		||	||	||	||				
9		120,50	49,50	189,20	110,80				
10		||	||	||	||				
11	b_j	120,50	49,50	189,20	110,80				
13		7,00	8,00	1,00	2,00				
14	$[c_{ij}]$ =	4,00	5,00	9,00	8,00				
15		9,00	2,00	3,00	6,00				
17		0,00	0,00	59,20	221,60				
18	$[c_{ij}x_{ij}]$ =	482,00	0,00	0,00	0,00		1251,80	←	Ymin
19		0,00	99,00	390,00	0,00				

Рис. 1.27 – Екран з проміжними і кінцевими результатами

Оптимальним рішенням завдання є матриця значень змінних $\mathbf{X}^* = \begin{bmatrix} 0{,}0 & 0{,}0 & 59{,}2 & 110{,}8 \\ 120{,}5 & 0{,}0 & 0{,}0 & 0{,}0 \\ 0{,}0 & 49{,}5 & 130{,}0 & 0{,}0 \end{bmatrix}$, що забезпечує мінімальне значення цільової функції $y^* = 1251{,}8$ *ум. од.*

Як бачимо, розв’язання відкритої задачі (збільшення запасів сировини при інших рівних умовах) приводить до кращого результату. Витрати на перевезення в порівнянні з попереднім прикладом скоротилися на 75,8 *ум. од*.

1.4.3. Транспортна задача із цілочисловою математичною моделлю

Змістовна постановка задачі

Загальна постановка транспортної задачі організації перевезень із цілочисловою математичною моделлю полягає у визначенні оптимального плану перевезень деякого однорідного вантажу з m пунктів виробництва $A_1, A_2, \ldots, A_m$ у n пунктів споживання $B_1, B_2, \ldots, B_n$. Для кожного пункту відправлення задані об'єми виробництва $a_1, a_2, \ldots, a_m$. Для кожного пункту призначення задані об'єми споживання $b_1, b_2, \ldots, b_n$. Об'єми виробництва, споживання та перевезення вимірюються тільки в цілих одиницях. Потрібно скласти такий план перевезень, що цілком забезпечив би потреби всіх споживачів при мінімальних витратах на перевезення. При цьому, як і в транспортній задачі із неперервною моделлю, в якості критерію оптимальності зазвичай береться або мінімальна вартість перевезень усього вантажу, або мінімальний час його доставки, або мінімальний сумарний пробіг вантажного транспорту.

В загальному випадку закрита транспортна задача немає цілочислового рішення, тому вважається, що загальний об'єм виробництва в пунктах відправлення перевищує загальний об'єм споживання в пунктах призначення.

Математична постановка задачі

Уведемо позначення, аналогічні позначенням для неперервної моделі, а саме:

- c_{ij} – витрати на перевезення одиниці вантажу з i–го пункту відправлення в j–й пункт призначення;
- a_i – об'єм вантажу в i–му пункті відправлення;
- b_j – потреба у вантажі в j–му пункті призначення;
- x_{ij} – кількість одиниць вантажу, що перевозиться з i–го пункту відправлення в j–й пункт призначення.

Тоді цілочислова математична модель транспортної задачі має вигляд:

$$y = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij} \to \min_{x_{ij} \in \mathbf{\Omega} \subset \mathbf{Z}^n}, \tag{1.59}$$

$$\Omega: \quad f_i = \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = \overline{1,m}, \tag{1.60}$$

$$f_{m+j} = \sum_{i=1}^{m} x_{ij} \ge b_j, \quad j = \overline{1,n}, \tag{1.61}$$

$$x_{ij} \ge 0, \quad i = \overline{1,m}, \quad j = \overline{1,n}, \tag{1.62}$$

$$x_{ij} = \text{int}, \quad i = \overline{1,m}, \quad j = \overline{1,n}. \tag{1.63}$$

Тут $y, \ f_1, f_2, \dots, f_{m+n}$ – дискретні лінійні функції, що задані на невід'ємному октанті простору цілих чисел $\mathbf{Z}^n$. Такі функції, як правило, мають місце, коли вантаж являє собою лічену множину досить великих заготівель або комплектуючих, неподільних виробів, упаковки сипучих матеріалів і т. ін. Одиницями виміру об'єму такого вантажу можуть бути такі одиниці як штуки, упаковки, партії й т. ін., тобто одиниці, що не можуть дробитися.

Цілочислова математична модель транспортного задачі відрізняється від раніше розглянутої неперервної відкритої моделі (1.53), (1.55), (1.56), (1.58), насамперед, наявністю додаткової системи обмежень на цілочислові змінні x_{ij}, тобто наявністю системи (1.63). Крім того, системи рівностей (1.55) заміщується системою нерівностей (1.61). Пояснюється це тим, що у випадку, коли потреби у вантажу визначаються дрібними числами задовольнити цілочислові обмеження у вигляді системи рівностей (1.55) принципово неможливо. Тому за рішенням задачі (1.59) – (1.63) кожний пункт призначення з дрібними потребами b_j, $j \in \left\{\overline{1,m}\right\}$, буде вимушено отримувати зайвий вантаж, що збільшить дрібні потреби до найближчої цілочислової величини.

Приклад транспортної задачі із цілочисловою математичною моделлю

За приклад візьмемо задачу з підрозділу 1.4.1, що розглядалась як задача з неперервною відкритою моделлю. Однак додамо в умову задачі фразу про те, що сировина може транспортуватися тільки в упаковках і що вага сировини в упаковці становить 1 *од.* Така зміна умови перетворює раніше розглянуту задачу із неперервною відкритою математичною моделлю в задачу із цілочисловою моделлю.

Зауважимо також, що введення умови (1.63) різко скорочує область припустимих рішень, перетворюючи його з неперервної замкнутої множини точок у лічену обмежену множину точок. Як правило, за інших рівних умов таке перетворення приводить до погіршення результатів оптимізації. В кращому випадку ці результати співпадають.

Математична і цифрова моделі транспортної задачі із закритою математичною моделлю в умовах прикладу

Математична модель задачі з урахуванням цілочислового простору змінних $\mathbf{R}^n$ буде незначно відрізнятися від математичної моделі з неперервним простором $\mathbf{E}^n$. В умовах прикладу ці зміни стосуються появb додатковій системи обмежень (1.63) і заміни системи рівностей (1.55) на систему нерівностей (1.61). В умовах прикладу математична модель набуває вигляду:

$$y = 7x_{11} + 8x_{12} + x_{13} + 2x_{14} + 4x_{21} + 5x_{22} + 9x_{23} +$$

$$+ 8x_{24} + 9x_{31} + 2x_{32} + 3x_{33} + 6x_{34} \to \min_{x_{ij} \in \boldsymbol{\Omega}} ,$$

$$\boldsymbol{\Omega}: \quad f_1 = x_{11} + x_{12} + x_{13} + x_{14} \le 170 ,$$

$$f_2 = x_{21} + x_{22} + x_{23} + x_{24} \le 150 ,$$

$$f_3 = x_{31} + x_{32} + x_{33} + x_{34} \le 180 ,$$

$$f_4 = x_{11} + x_{21} + x_{31} \ge 120.5 ,$$

$$f_5 = x_{12} + x_{22} + x_{32} \ge 49.5 ,$$

$$f_6 = x_{13} + x_{23} + x_{33} \ge 189.2 ,$$

$$f_7 = x_{14} + x_{24} + x_{34} \ge 110.8 ,$$

$$x_{ij} \ge 0; \quad i = \overline{1,3}; \quad j = \overline{1,4} ,$$

$$x_{ij} = \text{int}; \quad i = \overline{1,3}; \quad j = \overline{1,4} .$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Цифрова модель задачі теж матиме незначні зміни, а саме: масиви клітинок B5:E7, B9:E9, B11:E11, G5:G7 і I5:I7 тепер будуть мати *числовий* формат із кількістю десяткових знаків, рівним 0. Формат інших клітинок залишається таким самим, тобто з кількістю десяткових знаків, рівним 2.

Після завантаження в клітинки всіх констант і формул для обчислення проміжних і кінцевого результатів треба виконати необхідні установки даних у діалоговому вікні команди *Сервіс/Пошук_рішення* (рис. 1.28).

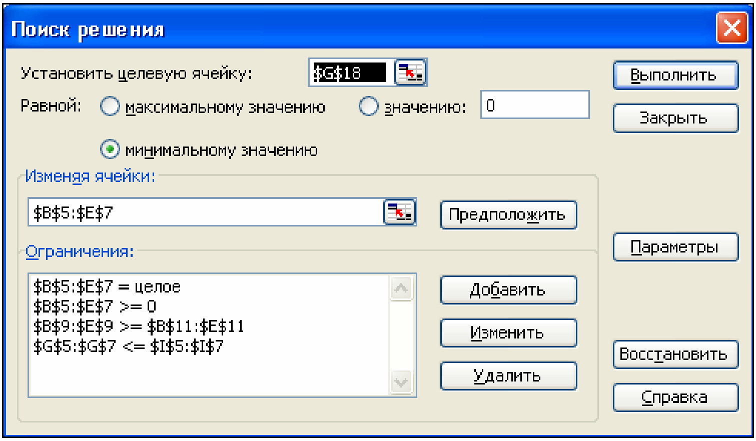


Рис. 1.28 – Діалогове вікно команди *Сервіс/Пошук_рішення* після установки даних

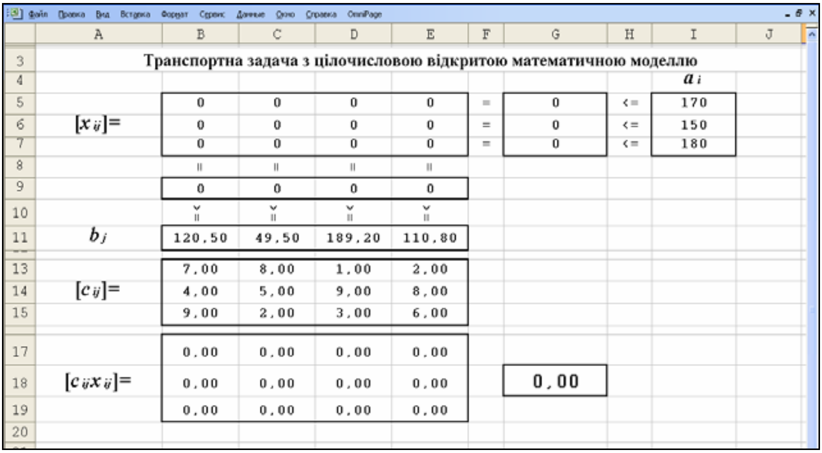
Як видно з рис. 1.28, установки в діалоговому вікні в порівнянні з попередніми прикладами перетерпіли зміни. Тепер вони такі:

- цільова клітинка – **$G$18**;

- тип екстремуму – **мінімум**;
- клітинки для змінних задачі – **$B$5:$E$7**;
- обмеження: **$B$5:$E$7 = ціле**;
  $B$5:$E$7 >= 0;
  $B$9:$E$9 >= $B$11:$E$11;
  $G$5:$G$7 <= $I$5:$I$7.

Установка обмеження **$B$5:$E$7 = ціле** при відпрацьовуванні команди *Сервіс/Пошук_рішення* приводить до виконання обчислювальних операцій тільки з цілими значеннями змінних.

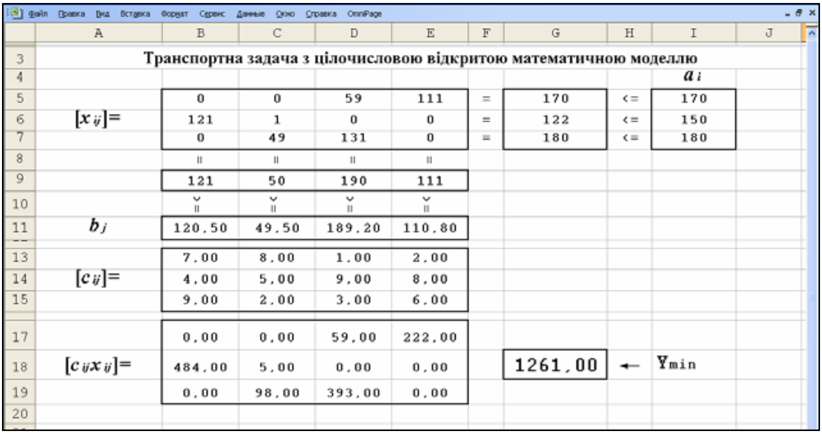
На рис.1.29 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.



	A	B	C	D	E	F	G	H	I	J
3	Транспортна задача з цілочисловою відкритою математичною моделлю									
4									a_i	
5		0	0	0	0	=	0	<=	170	
6	$[x_{ij}]=$	0	0	0	0	=	0	<=	150	
7		0	0	0	0	=	0	<=	180	
8		=	=	=	=					
9		0	0	0	0					
10		>=	>=	>=	>=					
11	b_j	120,50	49,50	189,20	110,80					
13		7,00	8,00	1,00	2,00					
14	$[c_{ij}]=$	4,00	5,00	9,00	8,00					
15		9,00	2,00	3,00	6,00					
17		0,00	0,00	0,00	0,00					
18	$[c_{ij}x_{ij}]=$	0,00	0,00	0,00	0,00		0,00			
19		0,00	0,00	0,00	0,00					
20										

Рис. 1.29 – Екран із вихідними установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.30.



	A	B	C	D	E	F	G	H	I	J
3	Транспортна задача з цілочисловою відкритою математичною моделлю									
4									a_i	
5		0	0	59	111	=	170	<=	170	
6	$[x_{ij}]=$	121	1	0	0	=	122	<=	150	
7		0	49	131	0	=	180	<=	180	
8		॥	॥	॥	॥					
9		121	50	190	111					
10		≥	≥	≥	≥					
11	b_j	120.50	49.50	189.20	110.80					
13		7.00	8.00	1.00	2.00					
14	$[c_{ij}]=$	4.00	5.00	9.00	8.00					
15		9.00	2.00	3.00	6.00					
17		0.00	0.00	59.00	222.00					
18	$[c_{ij}x_{ij}]=$	484.00	5.00	0.00	0.00		1261.00	←	Ymin	
19		0.00	98.00	393.00	0.00					
20										

Рис. 1.30 – Екран з проміжними і шуканими результатами

Оптимальним рішенням задачі є матриця цілих значень змінних $\mathbf{X}^* = \begin{bmatrix} 0 & 0 & 59 & 111 \\ 121 & 1 & 0 & 0 \\ 0 & 49 & 131 & 0 \end{bmatrix}$, що забезпечує мінімальне значення цільової функції $y^* = 1261$ *ум. од.*.

Порівняльний аналіз отриманого рішення з рішенням у попередньому прикладі показує, що введення обмеження на цілі значення для змінних спричинило погіршення цільової функції на 9,2 *ум. од.*(раніше було 1251,2 *ум.од.*). Крім того, пункти призначення отримали зайвий вантаж.

1.4.4. Транспортна задача про перевезення вантажу за два етапи

Змістовна постановка задачі

Однорідний вантаж потрібно доставити з *m* пунктів відправлення в *n* пунктів призначення. При доставці в пункти призначення вантаж може бути спочатку доставлений на *p* перевалочні пункти. Задано вартості перевезень із кожного пункту відправлення в кожний пункт призначення s_{ij} $(i=\overline{1,m},\quad j=\overline{1,n}\ )$ і перевалочний пункт d_{ik} $(i=\overline{1,m},\quad k=\overline{1,p}\ )$, а також вартості перевезення з кожного перевалочного пункту в кожний пункт призначення h_{kj} $(k=\overline{1,p},\quad j=\overline{1,n}\ )$. Потрібно скласти найбільш економічний план перевезення вантажу, що задовольняє потреби у вантажу в пунктах призначення. При цьому за завершенням перевезень товар повинний бути цілком вивезений із перевалочних пунктів.

Математична постановка задачі

Позначимо:

- a_i – запаси вантажу в *i*-му пункті відправлення;
- b_j – потреба у вантажі в *j*-му пункті призначення;
- c_k – місткість *k*-го перевалочного пункту;
- s_{ij} – вартість перевезення одиниці вантажу з *i*-го пункту відправлення в *j*-й пункт призначення, $i=\overline{1,m},\ \ j=\overline{1,n}$;
- d_{ik} – вартість перевезення одиниці вантажу з *i*-го пункту відправлення в *k*-й перевалочний пункт $i=\overline{1,m},\ \ k=\overline{1,p}$;
- h_{kj} – вартість перевезення одиниці вантажу з *k*-го перевалочного пункту в *j*-й пункт призначення $k=\overline{1,p},\ \ j=\overline{1,n}$;
- x_{ij} – кількість вантажу, що перевозиться з *i*-го пункту відправлення в *j*-й пункт призначення;
- y_{ik} – кількість вантажу, що перевозиться з *i*-го пункту відправлення в *k*-й перевалочний пункт;
- z_{kj} – кількість вантажу, що перевозиться з *k*-го перевалочного пункту в *j*-й пункт призначення.

Математична модель задачі з урахуванням вище наведених позначень може бути подана у вигляді задачі лінійного програмування:

$$w = \sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij} \cdot x_{ij} + \sum_{i=1}^{m}\sum_{k=1}^{p} d_{ik} \cdot y_{ik} + \sum_{k=1}^{p}\sum_{j=1}^{n} h_{kj} \cdot z_{kj} \to \min_{x_{ij}, y_{ik}, z_{kj} \in \Omega}, \tag{1.64}$$

$$\Omega: \ \sum_{j=1}^{n} x_{ij} + \sum_{k=1}^{p} y_{ik} \le a_i, \quad i = \overline{1,m}, \tag{1.65}$$

$$\sum_{i=1}^{m} x_{ij} + \sum_{k=1}^{p} z_{kj} \ge b_j, \quad j = \overline{1,n}, \tag{1.66}$$

$$\sum_{i=1}^{m} y_{ik} \le c_k, \quad k = \overline{1,p}, \tag{1.67}$$

$$\sum_{i=1}^{m} y_{ik} = \sum_{j=1}^{n} z_{kj}, \quad k = \overline{1,p}, \tag{1.68}$$

$$x_{ij} \ge 0, \ y_{ik} \ge 0, \ z_{kj} \ge 0, \ i = \overline{1,m}, \ j = \overline{1,n}, \ k = \overline{1,p}. \tag{1.69}$$

Тут цільова функція (1.64) визначає сумарні грошові витрати на перевезення вантажу в пункти призначення за два етапи, тобто з використанням перевалочних пунктів. Умова (1.65) обмежує об'єми перевезеного вантажу можливостями заводів-виготовлювачів. Обмеження (1.66) ставить за обов'язок компанії задовольнити потреби всіх баз у товарі. Обмеження (1.67) виключає порушення умови місткості складів. Вираз (1.68) вимагає повного звільнення складів після завершення перевезень. Нерівності (1.69) визначають простір можливих значень змінних.

Приклад транспортної задачі про перевезення вантажу за два етапи

У склад компанії входять 2 заводи та 4 склади. Склади мають місткість 45000, 20000, 30000, 15000 *од.* продукції, відповідно. Компанія забезпечує товарами п'ять торгових баз. Запити споживачів складають 30000, 23000, 15000, 32000, 16000 *од.* продукції, відповідно. Задача компанії – мінімізація витрат на перевезення товарів від заводів до складів, від заводів до баз і від складів до баз. Витрати на перевезення одиниці продукції від виготовлювачів на бази і склади, а також із складів на бази зведені в табл.1.2. Кількість товару, отриманого складом із заводів, повинно збігатися з кількістю товару, що вивозиться зі складу на бази. Як спланувати перевезення товарів, якщо заводи можуть зробити не більш 60 000 од. продукції кожний?

Таблиця 1.2 – Тарифи перевезень одиниці продукції

Відправники	Одержувачі				
	База 1	База 2	База 3	База 4	База 5
Завод 1	1.75	2.50	1.50	2.00	1.50
Завод 2	2.00	2.50	2.50	1.50	1.00
	Склад 1	Склад 2	Склад 3	Склад 4	
Завод 1	0.50	0.50	1.00	0.20	
Завод 2	1.50	0.30	0.50	0.20	
	База 1	База 2	База 3	База 4	База 5
Склад 1	1.50	1.50	0.50	1.50	3.00
Склад 2	1.00	0.50	0.50	1.00	0.50
Склад 3	1.00	1.50	2.00	2.00	0.50
Склад 4	2.50	1.50	0.20	1.50	0.50

Математична модель транспортної задачі про перевезення вантажу за два етапи в умовах прикладу

Математична модель задачі при використанні позначень, прийнятих для загальної моделі про розвезення вантажу (1.64) – (1.69), буде мати вигляд:

$$w = 1{,}75x_{11} + 2{,}5x_{12} + 1{,}5x_{13} + 2x_{14} + 1{,}5x_{15} +$$

$$+ 2x_{21} + 2{,}5x_{22} + 2{,}5x_{23} + 1{,}5x_{24} + x_{25} +$$

$$+ 0{,}5y_{11} + 0{,}5y_{12} + y_{13} + 0{,}2y_{14} +$$

$$+ 1{,}5y_{21} + 0{,}3y_{22} + 0{,}5y_{23} + 0{,}2y_{24} +$$

$$+ 1{,}5z_{11} + 1{,}5z_{12} + 0{,}5z_{13} + 1{,}5z_{14} + 3z_{15} +$$

$$+ z_{21} + 0{,}5z_{22} + 0{,}5z_{23} + z_{24} + 0{,}5z_{25} +$$

$$+ z_{31} + 1{,}5z_{32} + 2z_{33} + 2z_{34} + 0{,}5z_{35} +$$

$$+ 2{,}5z_{41} + 1{,}5z_{42} + 0{,}2z_{43} + 1{,}5z_{44} + 0{,}5z_{45} \rightarrow \min_{x_j \in \mathbf{\Omega}} ,$$

$$\mathbf{\Omega} : f_1 = x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + y_{11} + y_{12} + y_{13} + y_{14} \le 60000,$$

$$f_2 = x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + y_{21} + y_{22} + y_{23} + y_{24} \le 60000,$$

$$f_3 = x_{11} + x_{21} + z_{11} + z_{21} + z_{31} + z_{41} \ge 30000,$$

$$f_4 = x_{12} + x_{22} + z_{12} + z_{22} + z_{32} + z_{42} \ge 23000,$$

$$f_5 = x_{13} + x_{23} + z_{13} + z_{23} + z_{33} + z_{43} \ge 15000,$$

$$f_6 = x_{14} + x_{24} + z_{14} + z_{24} + z_{34} + z_{44} \ge 32000,$$

$$f_7 = x_{15} + x_{25} + z_{15} + z_{25} + z_{35} + z_{45} \ge 16000,$$

$$f_8 = y_{11} + y_{21} \le 45000,$$

$$f_9 = y_{12} + y_{22} \le 20000,$$

$$f_{10} = y_{13} + y_{23} \le 30000,$$

$$f_{11} = y_{14} + y_{24} \le 15000,$$

$$f_{12} = y_{11} + y_{21} = z_{11} + z_{12} + z_{13} + z_{14} + z_{15},$$

$$f_{13} = y_{12} + y_{22} = z_{21} + z_{22} + z_{23} + z_{24} + z_{25},$$

$$f_{14} = y_{13} + y_{23} = z_{31} + z_{32} + z_{33} + z_{34} + z_{35},$$

$$f_{15} = y_{14} + y_{24} = z_{41} + z_{42} + z_{43} + z_{44} + z_{45},$$

$$x_{ij} \ge 0,\ y_{ik} \ge 0,\ z_{kj} \ge 0,\ i = \overline{1,2},\ j = \overline{1,5},\ k = \overline{1,4}.$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для транспортної задачі про перевезення вантажу в умовах прикладу можуть бути такими:

- клітинки B4:E5 – для задання тарифів перевезення, d_{ik}, $i = 1,2$, $k = \overline{1,4}$;
- клітинки B10:F11 – для задання тарифів перевезення c_{ij}, $i = 1,2$, $j = \overline{1,5}$;
- клітинки B16:F19 – для задання тарифів перевезення h_{kj}, $k = \overline{1,4}$, $j = \overline{1,5}$;

- клітинки H4:K5 – для шуканих змінних задачі y_{ik}, $i = 1,2, \quad k = \overline{1,4}$;

- клітинки H10:L11 – для шуканих змінних задачі x_{ij}, $i = 1,2, \quad j = \overline{1,5}$;

- клітинки H16:L19 – для шуканих змінних задачі z_{kj}, $k = \overline{1,4}, \quad j = \overline{1,5}$;

- клітинки N12:N13 – для задання виробничої потужності заводів a_i, $i = 1,2$;

- клітинки H7:K7 – для задання місткості складів c_k, $k = \overline{1,4}$;

- клітинки H21:L21 – для задання запитів у вантажу від баз b_j, $j = \overline{1,5}$;

- клітинки L4:L5 – для проміжних результатів із завантаженням формул для обчислення сум $\sum_{k=1}^{4} y_{ik}$, $i = 1,2$;

- клітинки H6:K6 – для проміжних результатів із завантаженням формул для обчислення сум $\sum_{i=1}^{2} y_{ik}$, $k = \overline{1,4}$;

- клітинки M10:M11 – для проміжних результатів із завантаженням формул для обчислення сум $\sum_{i=1}^{2} x_{ij}$, $j = \overline{1,5}$;

- клітинки M12:M13 – для обчислення функцій f_i, $i = 1,2$;

- клітинки M16:M19 – для проміжних результатів із завантаженням формул для обчислення сум $\sum_{j=1}^{5} z_{kj}$, $k = \overline{1,4}$;

- клітинки H20:L20 – для обчислення функцій f_{2+j}, $j = \overline{1,5}$;

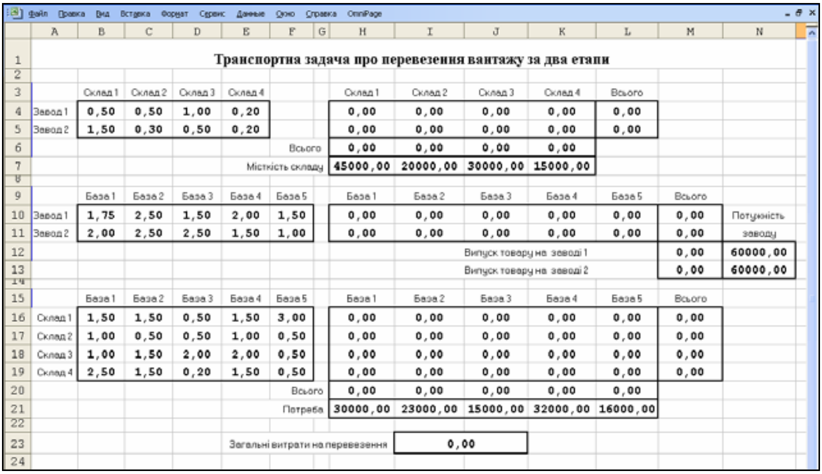
- клітинка I23 – для цільової функції *w* із відповідною формулою для її обчислення.

Всі клітинки електронної таблиці, що задіяні в розв'язанні задачі повинні мати *числовий* формат із числом десяткових знаків, рівним 2.

Після завантаження в електронну таблицю всіх необхідних констант і формул для обчислення проміжних і кінцевого результатів треба виконати необхідні установки даних у діалоговому вікні команди *Сервіс/Пошук_рішення*. В умовах приклада установки визначаються в такий спосіб:

- цільова клітинка – **$I$23**;
- тип екстремуму – **мінімум**;
- клітинки для змінних задачі – **$H$4:$K$5**; **$H$10:$L$11; $H$16:$L$19;**
- обмеження: **$H$10:$L$11 >= 0**;
  $H$16:$L$19 >= 0;
  $H$4:$K$5 >= 0;
  $H$20:$L$20 >= $H$21:$L$21;
  $H$6:$K$6 < = $H$7:$K$7;
  $H$6:$K$6 = $M$16:$M$19;
  $M$12:$M$13 <= $N$12:$N$13.

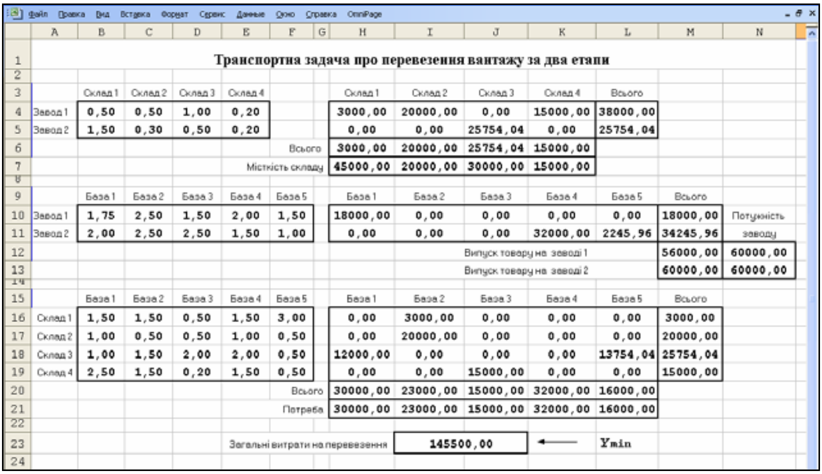
На рис.1.31 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1						Транспортна задача про перевезення вантажу за два етапи								
2														
3		Склад 1	Склад 2	Склад 3	Склад 4			Склад 1	Склад 2	Склад 3	Склад 4	Всього		
4	Завод 1	0,50	0,50	1,00	0,20			0,00	0,00	0,00	0,00	0,00		
5	Завод 2	1,50	0,30	0,50	0,20			0,00	0,00	0,00	0,00	0,00		
6						Всього		0,00	0,00	0,00	0,00			
7						Місткість складу		45000,00	20000,00	30000,00	15000,00			
8														
9		База 1	База 2	База 3	База 4	База 5		База 1	База 2	База 3	База 4	База 5	Всього	
10	Завод 1	1,75	2,50	1,50	2,00	1,50		0,00	0,00	0,00	0,00	0,00	0,00	Потужність
11	Завод 2	2,00	2,50	2,50	1,50	1,00		0,00	0,00	0,00	0,00	0,00	0,00	заводу
12										Випуск товару на заводі 1			0,00	60000,00
13										Випуск товару на заводі 2			0,00	60000,00
14														
15		База 1	База 2	База 3	База 4	База 5		База 1	База 2	База 3	База 4	База 5	Всього	
16	Склад 1	1,50	1,50	0,50	1,50	3,00		0,00	0,00	0,00	0,00	0,00	0,00	
17	Склад 2	1,00	0,50	0,50	1,00	0,50		0,00	0,00	0,00	0,00	0,00	0,00	
18	Склад 3	1,00	1,50	2,00	2,00	0,50		0,00	0,00	0,00	0,00	0,00	0,00	
19	Склад 4	2,50	1,50	0,20	1,50	0,50		0,00	0,00	0,00	0,00	0,00	0,00	
20						Всього		0,00	0,00	0,00	0,00	0,00		
21						Потреба		30000,00	23000,00	15000,00	32000,00	16000,00		
22														
23						Загальні витрати на перевезення			0,00					
24														

Рис. 1.31 – Екран із вихідними установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рі-шення* показаний на рис. 1.32.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1						Транспортна задача про перевезення вантажу за два етапи								
2														
3		Склад 1	Склад 2	Склад 3	Склад 4			Склад 1	Склад 2	Склад 3	Склад 4	Всього		
4	Завод 1	0,50	0,50	1,00	0,20			3000,00	20000,00	0,00	15000,00	38000,00		
5	Завод 2	1,50	0,30	0,50	0,20			0,00	0,00	25754,04	0,00	25754,04		
6						Всього		3000,00	20000,00	25754,04	15000,00			
7						Місткість складу		45000,00	20000,00	30000,00	15000,00			
8														
9		База 1	База 2	База 3	База 4	База 5		База 1	База 2	База 3	База 4	База 5	Всього	
10	Завод 1	1,75	2,50	1,50	2,00	1,50		18000,00	0,00	0,00	0,00	0,00	18000,00	Потужність
11	Завод 2	2,00	2,50	2,50	1,50	1,00		0,00	0,00	0,00	32000,00	2245,96	34245,96	заводу
12										Випуск товару на заводі 1			56000,00	60000,00
13										Випуск товару на заводі 2			60000,00	60000,00
14														
15		База 1	База 2	База 3	База 4	База 5		База 1	База 2	База 3	База 4	База 5	Всього	
16	Склад 1	1,50	1,50	0,50	1,50	3,00		0,00	3000,00	0,00	0,00	0,00	3000,00	
17	Склад 2	1,00	0,50	0,50	1,00	0,50		0,00	20000,00	0,00	0,00	0,00	20000,00	
18	Склад 3	1,00	1,50	2,00	2,00	0,50		12000,00	0,00	0,00	0,00	13754,04	25754,04	
19	Склад 4	2,50	1,50	0,20	1,50	0,50		0,00	0,00	15000,00	0,00	0,00	15000,00	
20						Всього		30000,00	23000,00	15000,00	32000,00	16000,00		
21						Потреба		30000,00	23000,00	15000,00	32000,00	16000,00		
22														
23						Загальні витрати на перевезення			145500,00		←	Ymin		
24														

Рис. 1.32 – Екран з проміжними і кінцевими результатами

Оптимальним рішенням задачі є матриці

$$\mathbf{X}^* = \begin{bmatrix} 18000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 32000 & 2246 \end{bmatrix},$$

$$\mathbf{Y}^* = \begin{bmatrix} 3000 & 20000 & 0 & 15000 \\ 0 & 0 & 25754 & 0 \end{bmatrix},$$

$$\mathbf{Z}^* = \begin{bmatrix} 0 & 3000 & 0 & 0 & 0 \\ 0 & 20000 & 0 & 0 & 0 \\ 12000 & 0 & 0 & 0 & 13754 \\ 0 & 0 & 15000 & 0 & 0 \end{bmatrix},$$

що забезпечують мінімальне значення цільової функції

$$w^* = 145500 \ \text{\textit{од.}}$$

1.4.5. Транспортна задача про перевезення вантажу декількох видів за два етапи

Змістовна постановка задачі

Вантаж, що включає q видів продукції, потрібно доставити з m пунктів відправлення в n пунктів призначення. При доставці в пункти призначення вантажі можуть бути спочатку доставлені на p перевалочних пунктів. Задано вартості перевезень для кожного виду вантажу з кожного пункту відправлення в кожний пункт призначення та кожний перевалочний пункт, а також вартості перевезення з кожного перевалочного пункту в кожний пункт призначення. Потрібно скласти найбільш економічний план перевезення вантажу, що задовольняє потреби у вантажу в пунктах призначення. При цьому товар за завершенням перевезень повинний бути цілком вивезений із перевалочних пунктів.

Математична модель задачі

Позначимо:

- a_{il} – запаси l-го виду вантажу в i-му пункті відправлення;
- b_{jl} – потреба в l-му виді вантажу в j-му пункті призначення;
- c_{kl} – місткість k-го перевалочного пункту стосовно l-го виду вантажу;
- c_{ijl} – вартість перевезення одиниці l-го виду вантажу з i-го пункту відправлення в j-й пункт призначення $i = \overline{1,m}$, $j = \overline{1,n}$, $l = \overline{1,q}$;
- c^{*}_{ikl} – вартість перевезення одиниці l-го виду вантажу з i-го пункту відправлення в k-й перевалочний пункт $i = \overline{1,m}$, $k = \overline{1,p}$, $l = \overline{1,q}$;
- c^{**}_{kjl} – вартість перевезення одиниці l-го виду вантажу з k-го перевалочного пункту в j-й пункт призначення $k = \overline{1,p}$, $j = \overline{1,n}$, $l = \overline{1,q}$;
- x_{ijl} – кількість l-го виду вантажу, що перевозиться з i-го пункту відправлення в j-й пункт призначення;
- y_{ikl} – кількість l-го виду вантажу, що перевозиться з i-го пункту відправлення в k-й перевалочний пункт;
- z_{kjl} – кількість l-го виду вантажу, що перевозиться з k-го перевалочного пункту в j-й пункт призначення.

Математична модель задачі з урахуванням вище наведених позначень може бути подана у вигляді задачі лінійного програмування:

$$w=\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q}c_{ijl}\cdot x_{ijl}+\sum_{i=1}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q}c_{ikl}^{*}\cdot y_{ikl}+$$

$$+\sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q}c_{kjl}^{**}\cdot z_{kjl}\to\min_{x_{ijl},y_{ikl},z_{kjl}\in\Omega}, \qquad (1.70)$$

$$\Omega:\quad \sum_{j=1}^{n}x_{ijl}+\sum_{k=1}^{p}y_{ikl}\le a_{il},\quad i=\overline{1,m},\quad l=\overline{1,q}, \qquad (1.71)$$

$$\sum_{i=1}^{m}x_{ijl}+\sum_{k=1}^{p}z_{kjl}\ge b_{jl},\quad j=\overline{1,n},\quad l=\overline{1,q}, \qquad (1.72)$$

$$\sum_{i=1}^{m}y_{ikl}\le c_{kl},\quad k=\overline{1,p},\quad l=\overline{1,q}, \qquad (1.73)$$

$$\sum_{i=1}^{m}y_{ikl}=\sum_{j=1}^{n}z_{kjl},\quad k=\overline{1,p},\quad l=\overline{1,q}, \qquad (1.74)$$

$$x_{ijl}\ge 0,\ y_{ikl}\ge 0,\ z_{kjl}\ge 0,\ i=\overline{1,m},\ j=\overline{1,n},\ k=\overline{1,p},\ l=\overline{1,q}. \qquad (1.75)$$

Через громіздкість математичних моделей конкретний приклад транспортної задачі про перевезення вантажу декількох видів за два етапи не наводиться.

1.4.6. Транспортна задача про перевезення вантажу декількох видів на запити споживачів за два етапи

Змістовна постановка задачі

Існує модифікація транспортної задачі про перевезення вантажу за два етапи, коли кількість вантажу в пунктах відправлення не фіксовано, а залежить від запитів споживачів.

Математична модель задачі

Позначимо:

- a_{il} – кількість l-го виду вантажу в i-му пункті відправлення;
- t_{il} – витрати на виробництво l-го виду вантажу в i-му пункті відправлення;
- b_{jl} – потреба в l-му виді вантажу в j-му пункті призначення;
- c_{kl} – місткість k-го перевалочного пункту стосовно l-го виду вантажу ;
- c_{ijl} – вартість перевезення одиниці l-го виду вантажу з i-го пункту відправлення до j-го пункту призначення $i=\overline{1,m},\ j=\overline{1,n},\ l=\overline{1,q}$;
- c^{*}_{ikl} – вартість перевезення одиниці l-го виду вантажу з i-го пункту відправлення в k-й перевалочний пункт $i=\overline{1,m},\ k=\overline{1,p},\ l=\overline{1,q}$;
- c^{**}_{kjl} – вартість перевезення одиниці l-го виду вантажу з k-го перевалочного пункту в j-й пункт призначення $k=\overline{1,p},\ j=\overline{1,n},\ l=\overline{1,q}$;
- b_{jl} – потреба в l-му виді вантажу в j-му пункті призначення;
- s_{il} – кількість виробленого l-го виду вантажу в i-му пункті відправлення;
- x_{ijl} – кількість l-го виду вантажу, що перевозиться з i-го пункту відправлення в j-й пункт призначення;
- y_{ikl} – кількість l-го виду вантажу, що перевозиться з i-го пункту відправлення в k-й перевалочний пункт;

- z_{kjl} – кількість l-го виду вантажу, що перевозиться з k-го перевалочного пункту до j-го пункту призначення.

Математична модель задачі з урахуванням вище наведених позначень може бути подана у вигляді задачі лінійного програмування:

$$w = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{q} c_{ijl} \cdot x_{ijl} + \sum_{i=1}^{m}\sum_{k=1}^{p}\sum_{l=1}^{q} c_{ikl}^{*} \cdot y_{ikl} +$$

$$+ \sum_{k=1}^{p}\sum_{j=1}^{n}\sum_{l=1}^{q} c_{kjl}^{**} \cdot z_{kjl} + \sum_{i=1}^{m}\sum_{l=1}^{q} t_{il} s_{il} \to \min_{x_{ijl}, y_{ikl}, z_{kjl}, s_{il} \in \Omega}, \quad (1.76)$$

$$\Omega: \quad \sum_{j=1}^{n} x_{ijl} + \sum_{k=1}^{p} y_{ikl} \le s_{il}, \quad i = \overline{1,m}, \quad l = \overline{1,q}, \quad (1.77)$$

$$\sum_{i=1}^{m} x_{ijl} + \sum_{k=1}^{p} z_{kjl} \ge b_{jl}, \quad j = \overline{1,n}, \ l = \overline{1,q}, \quad (1.78)$$

$$\sum_{i=1}^{m} y_{ikl} \le c_{kl}, \quad k = \overline{1,p}, \quad l = \overline{1,q}, \quad (1.79)$$

$$\sum_{i=1}^{m} y_{ik} = \sum_{j=1}^{n} z_{kj}, \quad k = \overline{1,p}, \ l = \overline{1,q}, \quad (1.80)$$

$$x_{ijl} \ge 0; \ y_{ikl} \ge 0; \ z_{kjl} \ge 0; \ s_{il} \ge 0; \ i = \overline{1,m}, \ j = \overline{1,n}, \ k = \overline{1,p},$$
$$l = \overline{1,q}. \quad (1.81)$$

Через громіздкість математичних моделей конкретний приклад транспортної задачі про перевезення вантажу декількох видів на запити споживачів за два етапи не наводиться.

1.4.7. Транспортна задача про закриття заводу

Змістовна постановка задачі

Виробниче об'єднання володіє m заводами та n складами. Задано потреби складів у продукції та вартості перевезення продукції з кожного заводу на кожний склад. Також задані фіксовані вартості функціонування заводів, задані можливості заводів стосовно виробництва продукції (потужність заводів). Виробниче об'єднання розглядає можливість закриття одного або декількох заводів з метою зменшити витрати на перевезення продукції. Які заводи, якщо це доцільно, повинні бути закриті?

Математична модель задачі

Позначимо:

- c_{ij} – вартості перевезення продукції з j-го заводу на i-й склад, $i = \overline{1,n}$, $j = \overline{1,m}$;
- d_i – потреби i-го складу в продукції, $i = \overline{1,n}$;
- a_j – можливість j-го заводу з виробництва продукції, $j = \overline{1,m}$;
- e_j – фіксована вартість функціонування j-го заводу, $j = \overline{1,m}$;
- z_j – двійкове число, що показує, чи потрібно закривати j-й завод (значення 0), чи ні (значення 1), $j = \overline{1,m}$;
- x_{ij} – кількість продукції, що перевозиться з j-го заводу на i-й склад, $i = \overline{1,n}$, $j = \overline{1,m}$.

Тоді математична модель транспортної задачі про закриття заводу може бути подана у вигляді:

$$y = \sum_{j=1}^{n}\left(e_j z_j + \sum_{i=1}^{m} c_{ij} x_{ij}\right) \to \min_{z_i, x_{ij} \in \Omega}, \qquad (1.82)$$

$$\Omega: \quad \sum_{i=1}^{n} x_{ij} \le z_j a_j, \quad j = \overline{1, m}, \qquad (1.83)$$

$$\sum_{j=1}^{m} x_{ij} \ge d_i, \quad i = \overline{1, n}, \qquad (1.84)$$

$$x_{ij} \ge 0, \quad i = \overline{1, n}; \quad j = \overline{1, m}, \qquad (1.85)$$

$$z_j \in \{0,1\}; \quad j = \overline{1, m}. \qquad (1.86)$$

Тут цільова функція (1.82) визначає загальні витрати виробничого об'єднання на функціонування заводів і транспортування готової продукції на склади. Обмеження (1.83) визначає можливості заводів стосовно виробництва продукції. Обмеження (1.84) визначають потреби складів у готовій продукції. Вимоги (1.85) і (1.86) визначають точковий простір змінних задачі.

Приклад транспортної задачі про закриття заводу

Виробниче об'єднання включає 5 заводів і 4 склади. Задано потреби складів у продукті, відповідно 15, 18, 14, 20 *од*, і матриця вартості на перевезення продукції з кожного заводу на кожний склад (в *ум. од*).

$$\left[c_{ij}\right] = \begin{bmatrix} 40 & 20 & 30 & 30 & 40 \\ 25 & 30 & 25 & 30 & 30 \\ 12 & 18 & 26 & 41 & 30 \\ 22 & 26 & 31 & 37 & 32 \end{bmatrix}.$$

Також задані фіксовані вартості функціонування заводів, відповідно 120, 150, 170, 160, 130 *ум. од.*, і можливості заводів з виробництва продукції, відповідно 20, 22, 17, 19, 18 *од.* Виробниче об'єднання розглядає можливість закриття одного або декількох заводів. Це повинно зменшити витрати на перевезення продукції. Які заводи, якщо це доцільно, повинні бути закриті?

Математична модель транспортної задачі про закриття заводу в умовах прикладу

Математична модель задачі при використанні позначень, прийнятих для загальної моделі (1.82) – (1.86), буде мати вигляд:

$$y = 120z_1 + 40x_{11} + 25x_{21} + 12x_{31} + 22x_{41} +$$

$$+150z_2 + 20x_{12} + 30x_{22} + 18x_{32} + 26x_{42} +$$

$$+170z_3 + 30x_{13} + 25x_{23} + 26x_{33} + 31x_{43} +$$

$$+160z_4 + 30x_{14} + 30x_{24} + 41x_{34} + 37x_{44} +$$

$$+130z_5 + 40x_{15} + 30x_{25} + 30x_{35} + 32x_{45} \to \min_{z_i, x_{ij} \in \mathbf{\Omega}},$$

$$\mathbf{\Omega}: \quad f_1 = x_{11} + x_{21} + x_{31} + x_{41} \le 20z_1,$$

$$f_2 = x_{12} + x_{22} + x_{32} + x_{42} \le 22z_2,$$

$$f_3 = x_{13} + x_{23} + x_{33} + x_{43} \le 17z_3,$$

$$f_4 = x_{14} + x_{24} + x_{34} + x_{44} \le 19z_4,$$

$$f_5 = x_{15} + x_{25} + x_{35} + x_{45} \le 18z_5,$$

$$f_6 = x_{11} + x_{12} + x_{13} + x_{14} + x_{15} \ge 15,$$

$$f_7 = x_{21} + x_{22} + x_{23} + x_{24} + x_{25} \ge 18,$$

$$f_8 = x_{31} + x_{32} + x_{33} + x_{34} + x_{35} \geq 14 \ ,$$

$$f_9 = x_{41} + x_{42} + x_{43} + x_{44} + x_{45} \geq 20 \ ,$$

$$x_{ij} \geq 0 \ , \quad i = \overline{1,4} \, , \quad j = \overline{1,5} \, ,$$

$$z_j \in \{0,1\} \, , \quad j = \overline{1,5} \, .$$

Цифрова модель і розв'язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для транспортної задачі про закриття заводу в умовах прикладу можуть бути такими:

- клітинки B5:F8 – для шуканих змінних задачі $x_{ij}, i = \overline{1,4}, \ \ j = \overline{1,5}$;
- клітинки B3:F3 – для шуканих змінних задачі z_j, $j = \overline{1,5}$;
- клітинки B10:F13 – для задання тарифів перевезення, c_{ij}, $i = \overline{1,4}, \ \ j = \overline{1,5}$;
- клітинки B14:F14 – для задання можливостей заводу a_j, $j = \overline{1,5}$;
- клітинки B15:F15 – для задання фіксованих вартостей функціонування заводу e_j, $j = \overline{1,5}$;
- клітинки B17:F17 – для обчислення функції f_j, $j = \overline{1,5}$, із завантаженою формулою для обчислення суми $\sum_{i=1}^{4} x_{ij}$;

- клітинки H5:H8 – для обчислення функції f_{5+i}, $i = \overline{1,4}$, із завантаженою формулою для обчислення суми $\sum_{j=1}^{5} x_{ij}$;
- клітинки B18:F18 – для проміжних результатів із завантаженою формулою для обчислення добутку $z_j a_j$, $j = \overline{1,5}$;
- клітинки B19:F19 – для проміжних результатів із завантаженою формулою для обчислення суми $\sum_{i=1}^{4} c_{ij} x_{ij}$, $j = \overline{1,5}$;
- клітинки B20:F20 – для проміжних результатів із завантаженою формулою для обчислення добутку $e_j z_j$, $j = \overline{1,5}$;
- клітинки B21:F21 – для проміжних результатів із завантаженою формулою для обчислення суми $e_j z_j + \sum_{i=1}^{4} c_{ij} x_{ij}$, $j = \overline{1,5}$;
- клітинки J5:J8 – для задання потреби складу в продукції d_i, $i = \overline{1,5}$
- клітинка H21 – для цільової функції y із відповідною завантаженою формулою для її обчислення.

Всі клітинки електронної таблиці, що задіяні в розв’язанні задачі повинні мати *числовий* формат із числом десяткових знаків, рівним 2. Виключення складають клітинки B3:F3, формат яких повинний відповідати двійковим числам.

Після завантаження в електронну таблицю всіх необхідних констант і формул для обчислення проміжних і кінцевого результатів треба виконати необхідні установки даних у діалоговому вікні команди *Сервіс/Пошук_рішення*. В умовах приклада установки визначаються в такий спосіб:

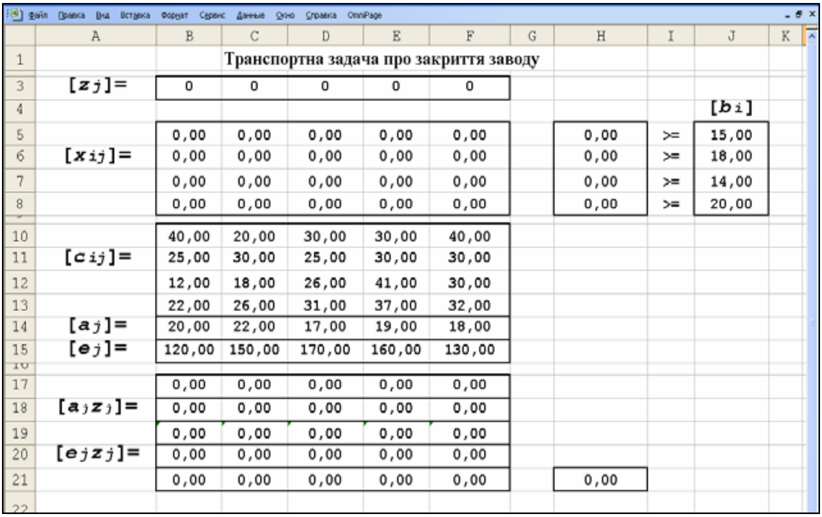
- цільова клітина – **$H$21**;
- тип екстремуму – **мінімум**;
- клітинки для змінних задачі: **$B$3:$F$3**; **$B$5:$F$8**;
- для обмеження: **$B$3:$F$3 = двійкове**;

  $B$5: $F$8 >= 0;

  $H$5:$H$8 >= $J$5:$J$8;

  $B$17:$F$17 <= $B$18:$F$18.

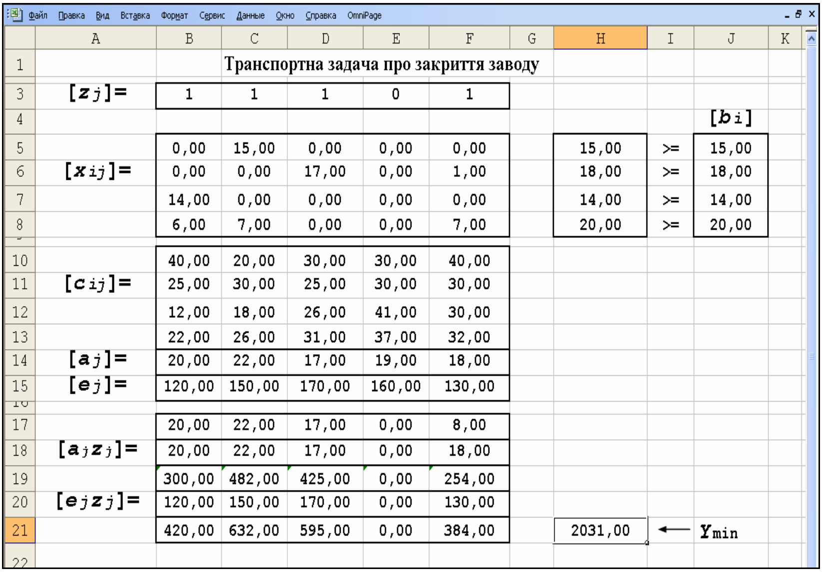
На рис.1.33 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.



	A	B	C	D	E	F	G	H	I	J	K
1		Транспортна задача про закриття заводу									
3	[zj]=	0	0	0	0	0					
4										[bi]	
5		0,00	0,00	0,00	0,00	0,00		0,00	>=	15,00	
6	[xij]=	0,00	0,00	0,00	0,00	0,00		0,00	>=	18,00	
7		0,00	0,00	0,00	0,00	0,00		0,00	>=	14,00	
8		0,00	0,00	0,00	0,00	0,00		0,00	>=	20,00	
10		40,00	20,00	30,00	30,00	40,00					
11	[cij]=	25,00	30,00	25,00	30,00	30,00					
12		12,00	18,00	26,00	41,00	30,00					
13		22,00	26,00	31,00	37,00	32,00					
14	[aj]=	20,00	22,00	17,00	19,00	18,00					
15	[ej]=	120,00	150,00	170,00	160,00	130,00					
17		0,00	0,00	0,00	0,00	0,00					
18	[ajzj]=	0,00	0,00	0,00	0,00	0,00					
19		0,00	0,00	0,00	0,00	0,00					
20	[ejzj]=	0,00	0,00	0,00	0,00	0,00					
21		0,00	0,00	0,00	0,00	0,00		0,00			
22											

Рис. 1.33 – Екран із вихідними установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.34.



	A	B	C	D	E	F	G	H	I	J	K
1		Транспортна задача про закриття заводу									
3	$[z_j]=$	1	1	1	0	1					
4										$[b_i]$	
5		0,00	15,00	0,00	0,00	0,00		15,00	>=	15,00	
6	$[x_{ij}]=$	0,00	0,00	17,00	0,00	1,00		18,00	>=	18,00	
7		14,00	0,00	0,00	0,00	0,00		14,00	>=	14,00	
8		6,00	7,00	0,00	0,00	7,00		20,00	>=	20,00	
10		40,00	20,00	30,00	30,00	40,00					
11	$[c_{ij}]=$	25,00	30,00	25,00	30,00	30,00					
12		12,00	18,00	26,00	41,00	30,00					
13		22,00	26,00	31,00	37,00	32,00					
14	$[a_j]=$	20,00	22,00	17,00	19,00	18,00					
15	$[e_j]=$	120,00	150,00	170,00	160,00	130,00					
17		20,00	22,00	17,00	0,00	8,00					
18	$[a_j z_j]=$	20,00	22,00	17,00	0,00	18,00					
19		300,00	482,00	425,00	0,00	254,00					
20	$[e_j z_j]=$	120,00	150,00	170,00	0,00	130,00					
21		420,00	632,00	595,00	0,00	384,00		2031,00	←	Y_{min}	
22											

Рис. 1.34 – Екрана з проміжними і кінцевими результатами

Оптимальним рішенням задачі є вектор-рядок

$$\mathbf{Z}^{*\mathrm{T}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \end{bmatrix} \text{ і матриця } \mathbf{Z}^{*} = \begin{bmatrix} 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 17 & 0 & 1 \\ 14 & 0 & 0 & 0 & 0 \\ 6 & 7 & 0 & 0 & 7 \end{bmatrix},$$

що забезпечують мінімальне значення цільової функції $y^{*} = 2031{,}0$ *ум. од.*

Аналіз розв’язку задачі дозволяє зробити висновок, що закриття 4-го заводу приведе до скорочення витрат на перевезення продукту без порушення будь яких інших умов на постачання продукту.

1.4.8. Транспортна задача про розиграш кубка

Змістовна постановка задачі

Спортивний кубок розігрують 2^n клубних команд ($n = 1,2,\ldots$). Тренувальні бази команд знаходяться в різних містах. Змагання проводяться за системою *play-off*. У кожному турі зустрічаються дві команди, одна з яких за результатом двох ігор («у гостях» і «удома») вибуває з подальшої боротьби. Відомі відстані між містами. Як організатори змагань повинні скласти пари команд на перший тур, щоб сумарна відстань при переїздах команд із міста в місто була мінімальною?

Математична модель задачі

Позначимо:

- l_{ij} – відстань між i-м і j-м містами, $i = \overline{1,2^n}$, $j = \overline{1,2^n}$;
- x_{ij} – змінна, що приймає значення 1, якщо i-а команда в турі грає з j-ю командою, і значення 0 – у протилежному випадку, $i = \overline{1,2^n}$, $j = \overline{1,2^n}$.

Математична модель задачі має вигляд:

$$y = 2\sum_{i=1}^{2^n-1}\sum_{j=i+1}^{2^n} l_{ij}x_{ij} \to \min_{x_{ij}\in\Omega}, \tag{1.87}$$

$$\Omega: \quad \sum_{\substack{i=1,\\ i\neq j}}^{2^n} x_{ij} = 1, \quad j = \overline{1,2^n}, \tag{1.88}$$

$$\sum_{\substack{j=1,\\ j\neq i}}^{2^n} x_{ij} = 1, \quad i = \overline{1,2^n}, \tag{1.89}$$

$$x_{ij} \in \{0, 1\}, \ i = \overline{1,2^n}, \ \ j = \overline{1,2^n} \ . \qquad (1.90)$$

$$x_{ij} = x_{ji}, \ i = \overline{1,2^n}, \ \ j = \overline{1,2^n} \ . \qquad (1.91)$$

Тут цільова функція (1.87) визначає сумарні витрати на проведення змагань. При моделюванні витрат вважається, що відстань з пункту А в В дорівнює відстані від пункту В до А, а також те, що команда не може грати сама із собою. Обмеження (1.88) говорить про те, що *j*-а команда повинна обов'язково зіграти з однією із інших команд. Аналогічно обмеження (1.89) говорить про те, що *i*-а команда теж повинна прийняти участь в першому турі змагань. Вираз (1.90) визначає можливі значення змінних. Рівність (1.91) змушує *j*-у команду зустрічається тільки з *i*-ю, якщо *i*-а команда їде на зустріч з *j*-ю, і навпаки.

Приклад транспортної задачі про розиграш кубка

У черговому турі розиграшу спортивного кубка беруть участь 8 команд. Змагання проводяться за системою *play-off*. Команда, що вибуває зі змагань, визначається за результатами двох ігор: «у гостях» і «вдома». Задано матрицю відстаней у *км* між містами-учасниками туру:

$$\mathbf{L} = \begin{Bmatrix} 0 & 150 & 400 & 370 & 120 & 600 & 350 & 200 \\ 155 & 0 & 180 & 500 & 420 & 280 & 220 & 530 \\ 400 & 180 & 0 & 370 & 450 & 280 & 310 & 440 \\ 370 & 500 & 370 & 0 & 300 & 220 & 360 & 530 \\ 120 & 420 & 450 & 300 & 0 & 140 & 270 & 540 \\ 600 & 280 & 280 & 220 & 140 & 0 & 390 & 170 \\ 350 & 220 & 310 & 360 & 270 & 390 & 0 & 270 \\ 200 & 530 & 440 & 530 & 540 & 170 & 270 & 0 \end{Bmatrix} .$$

Розбити команди по парах для участі в іграх чергового туру так, щоб сумарна відстань переїзду команд для проведення всіх ігор туру була мінімальною.

Математична модель транспортної задачі про розиграш кубка в умовах прикладу

Математична модель задачі при використанні позначень, прийнятих для загальної моделі задача про розиграш спортивного кубка (1.87) – (1.91), буде мати вигляд:

$$y = 2\big(150x_{12} + 400x_{13} + 370x_{14} + 120x_{15} + 600x_{16} + 350x_{17} +$$
$$+ 200x_{18} + 180x_{23} + 500x_{24} + 420x_{25} + 280x_{26} + 220x_{27} +$$
$$+ 530x_{28} + 370x_{34} + 450x_{35} + 280x_{36} + 310x_{37} + 440x_{38} +$$
$$+ 300x_{45} + 220x_{46} + 360x_{47} + 530x_{48} + 140x_{56} + 270x_{57} +$$
$$+ 540x_{58} + 390x_{67} + 170x_{68} + 270x_{78}\big) \rightarrow \min_{x_{ij} \in \boldsymbol{\Omega}} ,$$

$$\boldsymbol{\Omega}: \quad \sum_{\substack{i=1,\\ i \neq j}}^{8} x_{ij} = 1, \quad j = \overline{1,8} ,$$

$$\sum_{\substack{j=1,\\ j \neq i}}^{8} x_{ij} = 1, \quad i = \overline{1,8} ,$$

$$x_{ij} \in \{0, 1\}, \quad i = \overline{1,8}, \quad j = \overline{1,8}, \quad i \neq j ,$$

$$x_{ij} = x_{ji}, \quad i = \overline{1,8}, \quad j = \overline{1,8}, \quad i \neq j .$$

Цифрова модель і розв’язання задачі в інформаційному середовищі *Microsoft Excel*

Розподіл і призначення клітинок електронної таблиці для транспортної задачі про розиграш спортивного кубку в умовах приклада можуть бути такими:

- клітинки D4:J4, C5, E5:J5, C6:D6, F6:J6, C7:E7, G7:J7, C8:F8, H8:J8, C9:G9, I9:J9, C10:H10, J10, C11:I11 – для шуканих змінних задачі $x_{ij}, i=\overline{1,8},\ \ j=\overline{1,8},\ \ i\neq j$;

- клітинки C13:J13 – для проміжних результатів із завантаженням формул для обчислення сум $\sum_{\substack{i=1,\\ i\neq j}}^{8} x_{ij}=1,\ \ j=\overline{1,8}$;

- клітинки L4:L11 – для проміжних результатів із завантаженням формул для обчислення суми $\sum_{\substack{j=1,\\ j\neq i}}^{8} x_{ij}=1, i=\overline{1,8}$;

- клітинки D15:J15, E16:J16, F17:J17, G18:J18, H19:J19, I20:J20, J21 – для задання відстаней $l_{ij}, i=\overline{1,7},\ j=\overline{i+1,8}$;

- клітинки D23:J23, E24:J24, F25:J25, G26:J26, H27:J27, I28:J28, J29 – для проміжних результатів із завантаженням формули для обчислення добутку $l_{ij}x_{ij}, i=\overline{1,7},\ j=\overline{i+1,8}$;

- клітинки D31:J31 – для проміжних результатів із завантаженням формули для обчислення сум $\sum_{i=1}^{j-1} l_{ij}x_{ij},\ \ j=\overline{2,8}$;

- клітинка L31 – для цільової функції y із завантаженою відповідною формулою для її обчислення.

Всі клітинки електронної таблиці, що задіяні в розв’язанні задачі повинні мати *числовий* формат із числом десяткових знаків, рівним 0.

Після завантаження в електронну таблицю всіх необхідних констант і формул для обчислення проміжних і кінцевого результатів треба виконати необхідні установки даних у діалоговому вікні команди *Сервіс/Пошук_рішення*. В умовах прикладу установки визначаються в такий спосіб:

- цільова клітинка – **$L$31**;
- тип екстремуму – **мінімум**;
- клітинки для змінних задачі – **$D$4:$J$4**, **$C$5**, **$E$5:$J$5**, **$C$6:$D$6**, **$F$6:$J$6**, **$C$7:$E$7**, **$G$7:$J$7**, **$C$8:$F$8**, **$H$8:$J$8**, **$C$9:$G$9**, **$I$9:$J$9**, **$C$10:$H$10**, **$J$10**, **$C$11:$I$11**;
- обмеження: **$C$4:$J$11 = двійкове**;

$C$13: $J$13 = 1;

$L$4:$L$11 = 1;

$C$5 = $D$4;
$C$6 = $E$4;
. . .
$I$11 = $J$10.

$\left(2^n - n\right)$ рівностей для кожної змінної x_{ij}, $i = \overline{1,8}$, $j = \overline{1,8}$, $i \neq j$.

На рис.1.35 показаний вигляд екрана, що повинний передувати виконанню команди *Сервіс/Пошук_рішення*.

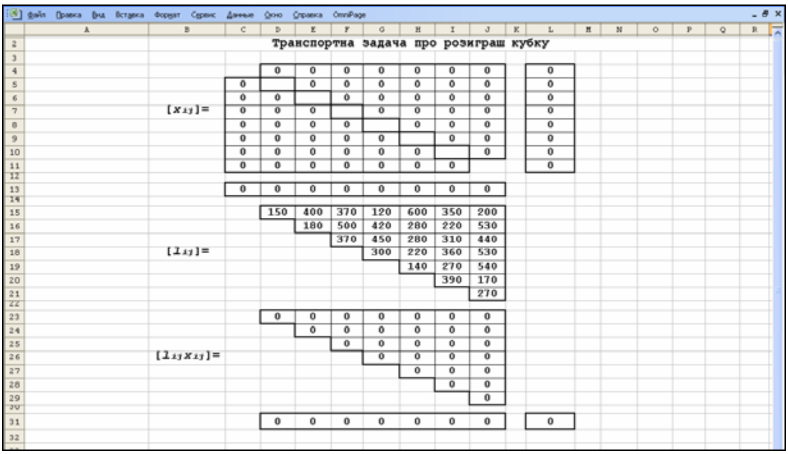


Рис. 1.35 – Екран із вихідними установками

Вигляд екрана після виконання команди *Сервіс/Пошук_рішення* показаний на рис. 1.36.

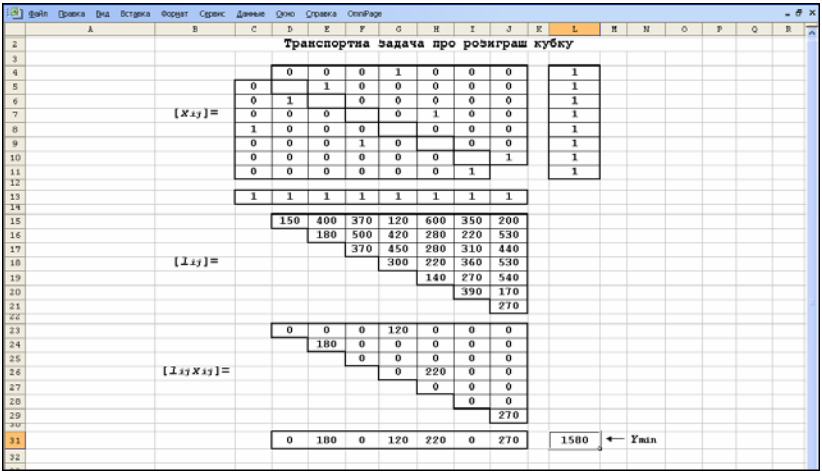


Рис. 1.36 – Екран з проміжними і кінцевими результатами

Оптимальним рішенням задачі є матриця

$$\mathbf{X}^* = \begin{bmatrix} \times & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \times & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \times & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \times & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & \times & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \times & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \times \end{bmatrix},$$

що забезпечує мінімальне значення цільової функції $y^* = 1580\,км.$

Примітка. У матриці $\mathbf{X}^*$ символом «×» позначені елементи, що відповідно до вимоги «$i \neq j$» не приймають участі в розв'язання задачі.

За результатами розв'язання оптимізаційної задачі випливає, що в черговому турі змагань повинні зустрітися такі пари команд: 1-а і 5-а; 2-а і 3-я; 4-а і 6-а, 7-а і 8-а. Такий розклад зустрічей забезпечує мінімальну сумарна відстань при переїздах команд, а саме, 1580 *км.*

* * *

Розглянуті математичні моделі не охоплюють всі можливі типи оптимізаційних транспортних задач. Безперечно, реальна практика перевезень буде постійно пропонувати нові задачі з більш складними математичними моделями. Класифікація задач, наведена на рис. 1.1, буде постійно розширюватися й удосконалюватися.

Основне завдання, що ставили перед собою автори – це, з одного боку, продемонструвати читачам найпростішу методику розв'язання оптимізаційних задач за допомогою убудованого в інформаційну систему *Microsoft Excel* програмного засобу «*Пошук рішення*». З іншого боку, показати на задачах малої вимірності адекватність моделей транспортних задач, яка у більшості випадків може бути перевірена прямим перебором можливих рішень.

За розглянутими розв'язаннями конкретних транспортних задач малої вимірності можна скласти уявлення про об'єм і складність робіт із вирішенням тих же задач великої вимірності. Незважаючи на те, що математичні моделі в задачах малої і великої вимірності майже збігаються, розв'язання останніх за допомогою розглянутої методики не завжди є можливим. Якщо оптимізаційні задачі малої вимірності можна вирішити навіть простим перебором і в реальному масштабі часу, то при вирішенні задач великої вимірності чинники часу, витрат динамічної пам'яті ЕОМ або складності підготовки вихідних даних починають грати суттєву роль і можуть стати причиною, що не дозволяє одержати рішення. Одним із можливих виходів із такого становища може бути розробка нових методів розв'язання, що нейтралізує вище перераховані чинники. При розробці нових методів іноді доводиться іти на компроміс, наприклад, пожертвувати точністю рішення заради економії часу обчислення.

Подальші розділи монографії будуть присвячені питанням розробки методів розв'язання транспортних задач великої вимірності, а також програмній реалізації цих задач та порівняльному аналізу з існуючими методами.

РОЗДІЛ 2

РОЗБИТТЯ ТРАНСПОРТНОЇ МЕРЕЖІ НА РАЙОНИ

Організація доставки дрібного вантажу у великих містах або великих регіонах із малонаселеними пунктами пов'язана з транспортною мережею, що об'єднує велику кількість споживачів і постачальників. Для таких мереж оптимізація процесу перевезень є складною й трудомісткою задачею.

З метою більш ефективної організації перевезень транспортну мережу розбивають на райони. При розбивці транспортної мережі на райони враховується карта місцевості, відстані між постачальниками і споживачами, час проїзду окремих ділянок дороги, об'єм постачань.

У даному дослідженні за критерій оцінки проведеного розбиття вибрана сумарна довжина маршрутів комівояжера за всіма районами розбиття. Альтернативним критерієм виступає сумарний час проїзду за всіма маршрутами.

2.1. Аналіз підходів до розв’язання задачі розбиття транспортної мережі на райони

Аналіз вимог, запропонованих до якості виконання послуги доставки, показує, що найбільш важливими є такі показники доставки як час, надійність, сумарні матеріальні витрати. У зв'язку з цим дуже актуальним є дослідження і розробка ефективних методів, що дозволяють планувати процес доставки з урахуванням перелічених показників транспортних послуг.

Постачальникам доводиться обслуговувати значну кількість клієнтів, у зв'язку з чим оптимізація процесу доставки виконується для транспортних мереж великої вимірності. Це ускладнює застосування деяких методів планування маршрутів, обмежує можливість побудови гнучкого графіка у випадку нерівномірності попиту, приводить до втрати клієнтів через несвоєчасність доставки.

Більш ефективної організації перевізного процесу можна досягти, застосувавши зонування адресів доставки, або розбиття транспортної мережі на райони. Згідно з думкою автора [7], який має великий досвід практичної роботи з організацією транспортних перевезень, розбиття на райони і додаткова фільтрація задач з перевезення вантажів за зонами доставки при комплектації рейсів дозволить уникнути невиправдано завищеного пробігу транспортних засобів за відсутністю алгоритмів оптимальної маршрутизації.

Питання розбиття транспортної мережі на райони (задача районування) на цей час широкого розвитку не одержало. Відомо лише декілька підходів до розбиття транспортної мережі на райони. Один із них [9] припускає використання в якості границь районів природних перешкод – рік, залізничних колій і т. ін. Даний підхід обумовлений припущеннями про те, що об'їзд природних перешкод тягне за собою значне збільшення загального пробігу. Пункти, що розділені такими перешкодами, вважаються розташованими на значній відстані один від одного. Однак в умовах міста даний підхід може не забезпечити можливості розбиття транспортної мережі на необхідну кількість районів. Крім того, отримані райони можуть суттєво відрізнятися один від одного як за розміром, так і за кількістю пунктів відправки та доставки вантажу.

Той же автор пропонує альтернативний варіант розбиття транспортної мережі, при якому транспортні райони збігаються з існуючими адміністративними районами. При цьому за границі районів можуть бути також використані центральні вулиці, жвавий рух на яких суттєво збільшує час, що витрачається на доставку усередині одного району. В розбитті, що отримане на цих засадах, у різних районах можуть опинитися близько розташовані один до одного пункти, включення яких в один район було б значно вигіднішим. Тому запропоновані в [9] критерії розбиття в загальному випадку не гарантують ефективного розбиття транспортної мережі на райони.

Досить ефективним може виявитися розбиття на райони, що виконується експертом з урахуванням карти місцевості та взаємного розташування пунктів доставки. Проте таке розбиття є сугубо суб'єктивним, а його формалізація – дуже важкою задачею, який би критерій при цьому не використовувався.

У [10] розбиття мережі на райони розглядається як частина розв’язання задачі розвезення, що вирішується за модифікованим методом Літтла-Кэрола-Мурті. В цьому випадку транспортна мережа динамічно розбивається на два мікрорайони, із яких тільки один є дійовим. Динамічний підхід забезпечує більш швидку роботу методу, однак не дозволяє одержувати базове розбиття всієї транспортної мережі. Такий підхід не дає можливості одержання фрагментів мереж, що надалі можна було б розглядати автономно.

У статті [6] викладений алгоритм розв’язання оптимізаційної задачі доставки дрібних вантажів в умовах великого міста методом локалізації. Локалізація, або, по суті, розбиття транспортної мережі міста на райони, розглядається як один з етапів процесу організації транспортних перевезень. Розв’язання задачі засновано на розподілі всіх клієнтів по групах за територіальною ознакою шляхом використання процедури кластерного аналізу на базі методу нечітких *c*-середніх. Реалізувати метод *c*-середніх пропонується за допомогою однієї з убудованих функцій пакету *MATHLAB*.

Метод нечітких *c*-середніх [11] призначений для розбивки множини елементів на нечіткі підмножини. Для кожного з елементів x_i визначається функція приналежності до *j*-ї множини u_{ij}, $0 \le u_{ij} \le 1$. Метод заснований на мінімізації цільової функції

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\|^2, \quad 1 \le m \le \infty \ , \tag{2.1}$$

де *m* – довільне дійсне число більше ніж 1, x_i – вектор даних, що характеризує положення *i*-го пункту; c_j – вектор даних, що характеризує положення центру *j*-го кластера, $\|*\|$ – квадратична норма для визначення зв'язку між центрами та іншими елементами.

Коригування функції приналежності робиться за формулою

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}} \ . \tag{2.2}$$

Дані про розташування центру коригуються відповідно до формули

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m} \ . \tag{2.3}$$

Ітеративний процес наближення до оптимального розбиття припиняється при виконанні умови $\max\limits_{i,j}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon$, де ε – задана точність рішення.

Стосовно до задачі локалізації пунктів транспортної мережі метод припускає наступну послідовність дій [6]. На підставі даних про взаємне розташування пунктів транспортної мережі вибираються центри кластерів (районів) транспортної мережі. Дані подаються у вигляді довільної кількості параметрів кожного пункту або ж у вигляді рисунка-карти з відзначеними на ній точками-пунктами (матриці вихідних даних, що відображають положення пунктів на карті). Відповідно до [6] кластерізація реалізується двома різними функціями в залежності від типу поданих даних.

Кожна точка розглядається як потенційний центр кластера. Для всіх точок обчислюється певна міра, що характеризує її здатність виступати як центр кластера. Міра конкретної точки в обов'язковому порядку залежить від щільності інших точок навколо неї.

При відомих параметрах пунктів алгоритм розбиття складається з двох повторюваних кроків:

1. Вибір центру кластера – точки з найбільшою мірою щільності.

2. Видалення з подальшого розбиття всіх точок в околі центру отриманого кластера та вибір центру наступного кластера. Радіус околу повинний бути заданий заздалегідь.

Циклічне повторення кроків відбувається доти, поки всі точки не опиняться всередині околу заданого радіуса. Оптимальна кількість кластерів визначається програмою в процесі розбиття.

При графічному завданні транспортної мережі кількість кластерів повинна бути вказана у вихідних даних до процедури. При пошуку рішення аналізується цільова функція.

Процес пошуку рішення також є ітеративним. Він припиняється при досягненні заданого числа ітерацій або за умови, що різниця значень цільової функції на поточній і попередній ітераціях не перевищує заданої точності.

Обидві функції як результат видають координати центрів кластерів і значення функції приналежності для кожної пари точка-кластер.

Перевагою запропонованого способу одержання районів є наявність відповідного програмного забезпечення. До того ж досить зручним є одержання результату у вигляді нечітких множин. Дана особливість дає можливість розподілу периферійних пунктів за районами, виходячи з додаткових умов.

У роботі [6] алгоритм охарактеризований як швидкодіючий. Однак авторами даної роботи розбиття на райони робилося тільки на підставі розташування пунктів транспортної мережі на площині – кожний пункт описувався парою координат по осях *X* і *У*. Особливості транспортних зв'язків між пунктами, природні перешкоди і т. ін. не були враховані. Пакет *MATLAB* дозволяє врахувати також і ці параметри, однак така задача зажадає додаткових досліджень і може суттєво знизити швидкодію алгоритму. Недоліком використання готового програмного пакета є складність внесення додаткових параметрів і обмежень.

У джерелі [12] метод кластерізації подається як метод із великими строковими витратами, за якими кожний додатковий елемент збільшує час вирішення задачі на цілий порядок.

У роботі [13] задача районування транспортної мережі формулюється як задача цілочислового лінійного програмування з цільовою функцією

$$\sum_{k=1}^{r}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{i,j} x_{i,k} x_{j,k} \to \min \tag{2.4}$$

при обмеженнях

$$\begin{cases} \sum_{k=1}^{r} x_{i,k} = 1, \quad \forall i \in \{1,...,n\}; \\ \sum_{i=1}^{n} d_i x_{i,k} \le w, \quad \forall k \in \{1,...,r\}; \\ x_{i,k} \in \{0,1\}\ , \ \forall i \in \{1,...,n\}, \quad \forall k \in \{1,...,r\}, \end{cases} \tag{2.5}$$

де r – загальна кількість підмножин розбиття, що дорівнює кількості наявних транспортних засобів; $c_{i,j}$ – відстань між *i*-м і *j*-м клієнтами; $x_{i,k}$ – двійкові змінні ($x_{i,k} = 1$, якщо *i*-й клієнт входить у *k*-у підмножину розбивки, і $x_{i,k} = 0$ – у протилежному випадку); d_i – кількість вантажу, що потрібно доставити *i*-му клієнту; *w* – максимально припустима сумарна вага замовлень усіх клієнтів у даній підмножині розбиття, що дорівнює вантажопідйомності транспортного засобу.

У даній постановці передбачається, що в результаті розв'язання задачі буде отримано розбиття транспортної мережі на райони з мінімальною довжиною маршрутів. При цьому сумарний об'єм замовлень клієнтів усередині одного району не буде перевищувати заздалегідь заданого числа w, рівного вантажопідйомності автомобіля. Перевагою запропонованого формулювання задачі розбиття транспортної мережі на райони є ємність математичної моделі, що враховує як пробіг транспортного засобу усередині кожного з районів, так і його вантажопідйомність. Недоліком є складність розв'язання задачі, що подана як задача цілочислового лінійного програмування (ЦЛП). Відповідно до досліджень авторів [14] про час, що витрачається на розв'язання задачі ЦЛП за допомогою спеціальних методів складно судити заздалегідь, дуже часто він не залежить від вимірності задачі. За прийнятний час можна гарантовано вирішити тільки задачі порівняно невеликої вимірності, що не є актуальним.

Таким чином, розроблені на даний момент системи рекомендацій не гарантують оптимального розбиття транспортної мережі на райони. Існуючі підходи або не формалізовані та не пристосовані до застосування ЕОМ, або не пристосовані для врахування всіх суттєвих чинників, або не пристосовані для розв’язання задач великої вимірності. Крім того, не існує критерію, що дозволяє оцінити ефективність розбиття транспортної мережі на райони. У зв'язку з цим дослідження задачі розбиття транспортної мережі на райони є, безумовно, необхідним і перспективним. Саме задача районування закладає базові чинники, що сприяють ефективному вирішенню інших задач, що безпосередньо пов’язані з процесом пасажирських або вантажних перевезень.

2.2. Методика проведення дослідження

Для розбиття транспортної мережі на райони розроблений спеціальний метод, що базується на відомому комбінаторному методі «гілок і границь». Нова модифікація методу з'явилася як результат багатовекторного дослідження одного із авторів цієї монографії, а саме – А. О. Кобець.

У процесі дослідження були розглянуті різні прийнятні варіанти модифікації методу «гілок і границь». Усі варіанти пройшли комплексне тестування, метою якого було встановити для кожного варіанта ступінь задоволення вибраному критерію якості розбиття.

Для оцінки якості розбиття була проведена серія обчислювальних експериментів. На початковому етапі різні варіанти модифікації пройшли тестування на задачах малої вимірності. Це були задачі розбиття на 2 або 3 райони для транспортних мереж, що містять від 8 до 18 пунктів доставки. Якість отриманого розбиття оцінювалася за критерієм сумарної довжини маршрутів комівояжера, які визначалися як результат розв’язання задачі комівояжера для кожного із районів окремо. Результати розбиття транспортної мережі на райони за модифікованим методом «гілок і границь» порівнювалися з результатами, отриманими методом прямого перебору всіх можливих варіантів розбиття, тобто з дійсно оптимальними результатами.

Для дослідження були використані пункти доставки, що розташовувалися на місцевості випадковим чином. Відстань між ними визначалися по прямій. Пари безпосередньо зв'язаних між собою пунктів доставки відбиралися також випадково.

Кількість пунктів у районах розбиття строго фіксувалась. У випадку розбиття на два райони вона дорівнювала одній другій від загальної кількості пунктів. У випадку розбиття на три райони – одній третій. Таке обмеження дозволило зменшити загальну кількість можливих варіантів перебору і забезпечити порівняння результатів розбиття з дійсно оптимальним варіантом.

Значно актуальнішою є задача одержання районів розбиття, що займають приблизно однакові площі, або ж районів розбиття, при об'їзді всіх пунктів яких буде потрібно проїхати приблизно однакові відстані. При такій постановці вимог до розмірів району повний перебір для відшукання оптимального варіанта можливий тільки для задач із дуже малою вимірністю, що не викликає особливого інтересу. Тому при розробці та аналізі нових модифікацій методу «гілок і границь» спочатку розглядалися алгоритми розбиття на райони з рівною кількістю пунктів доставки. Серед цих алгоритмів був відібраний найбільш прийнятний варіант, на підставі котрого і був розроблений остаточний варіант модифікованого методу «гілок і границь» стосовно задачі районування транспортної мережі.

Запропонована остаточна модифікація методу «гілок і границь» дозволяє формувати райони, що незначно відрізняються друг від друга довжиною маршрутів комівояжера.

Далі була проведена перевірка остаточного алгоритму розбиття для пунктів доставки, що випадковим способом нанесені на карту міста. Вимірність задачі – до 60 пунктів доставки. Для оцінки відстаней між пунктами доставки були визначені довжини найкоротших маршрутів, що з'єднують ці пункти. Отримане розбиття транспортної мережі на райони модифікованим методом «гілок і границь» було нанесено на карту, після чого оцінювалось на предмет реального застосування та практичної доцільності.

Потім обмеження на кількість пунктів у районах було замінено більш актуальною вимогою незначної різниці між довжинами маршрутів комівояжера в районах. Якість отриманого розбиття оцінювалась візуально після їх нанесення на карту.

2.3. Адаптація методу «гілок і границь» до розв’язання задачі розбиття транспортної мережі на райони

У цьому підрозділі розглядаються питання адаптації методу «гілок і границь» для розв’язання задачі розбиття транспортної мережі на райони.

Метод «гілок і границь» традиційно використовується для розв’язання задачі комівояжера. Відповідно до цього методу будується дерево. Вершинами дерева є підмножини маршрутів. Ці підмножини входять до множини всіх маршрутів транспортної мережі, що містять або не містять певні ланки цієї мережі.

Кореневою вершиною дерева є множина всіх маршрутів комівояжера даної мережі, тобто маршрутів, що починаються в деякому пункті h, проходять через всі пункти транспортної мережі рівно один раз і закінчуються знову у пункті h. Для кореневої вершини є інцидентними дві проміжкові вершини (k,l) та $(\overline{k,l})$. Вершина (k,l) – це підмножина маршрутів комівояжера, всі маршрути якої містять ланку транспортної мережі від пункту k до пункту l. Для отримання наступної за кореневою вершини дерева обирається та ланка, при вилученні якої з маршруту комівояжеру нижня границя його довжини зростає більше, ніж при вилученні будь-якої іншої ланки. Вершина $(\overline{k,l})$ – це підмножина маршрутів комівояжера, жоден з маршрутів якої не містить ланки (k,l).

Довільна вершина дерева (i,j) позначає підмножину маршрутів, які проходять вдовж ланки (i,j). Також ці маршрути проходять вдовж всіх ланок (s,t), $s \in \{1,2,...,N\}/\{i,j,t\}$, $t \in \{1,2,...,N\}/\{i,j,s\}$, такі що вершини (s,t) містяться між вершиною (i,j) та кореневою вершиною дерева. Якщо між кореневою вершиною та вершиною (i,j) є вершини виду $(\overline{p,r})$, то серед маршрутів підмножини немає таких, які проходили б вдовж ланки (p,r).

Кожній вершині, тобто кожній підмножині, ставиться у відповідність точна нижня границя довжини маршруту. Вершини дерева вибираються таким чином, щоб нижня границя довжин відповідних маршрутів була мінімальною, і цикл утворювали тільки ті маршрути, що проходять через усі вершини транспортної мережі. У побудованому дереві вибирається піддерево з мінімальною довжиною маршруту комівояжера.

Для побудови дерева й оцінки нижніх границь довжин маршрутів формується зведена матриця відстаней, і всі обчислення ведуться не на підставі вихідної матриці відстаней, а на базі зведеної матриці.

Пропонується на підставі зведеної матриці будувати декілька дерев у кількості, що дорівнює заданому числу районів у розбитті. І далі, усі пункти, що відповідають вершинам кожного з дерев, відносити до одного району.

Приведемо змістовну постановку задачі розбиття транспортної мережі на райони, що включає вимогу до сумірності довжин маршрутів різних районів.

Змістовна постановка задачі. Для транспортної мережі, що містить N пунктів прийому товару, задані відстані між пунктами з транспортним зв'язком. Необхідно розбити мережу на R районів із мінімальною сумарною довжиною кільцевих маршрутів, що визначаються як рішення задачі комівояжера для кожного із районів. При цьому довжина маршрутів кожного із районів не повинна суттєво відрізнятися від довжин маршрутів інших районів.

Розв'язання задачі районування транспортною мережі за модифікованим методом «гілок і границь». Формування районів за допомогою запропонованого методу (модифікованого методу «гілок і границь») умовно складається з декількох етапів.

Першій етап. На першому етапі початкові просторові дані транспортної мережі перетворюють до вигляду, зручного для наступного комп'ютерного розв'язання задачі районування. При цьому спочатку складається матриця відстаней **D** вимірності $N \times N$, де N − число пунктів транспортної мережі. Елемент d_{ij} цієї матриці дорівнює або відстані пункту i від пункту j у випадку, якщо пункти i і j мають безпосередній транспортний зв'язок між собою ($i, j \in \{1,2,...,N\}$, $i \neq j$), або довжині мінімального маршруту з пункту i у пункт j – у протилежному випадку. Довжини маршрутів визначаються за допомогою матричного методу розв’язання задачі про найкоротші відстані, що описаний в [5].

Суть матричного методу полягає в наступному. На першому кроці задається матриця $\mathbf{R}^{(1)} = \left\{r_{ik}^{(1)}\right\}_{N \times N}$ з елементами

$$r_{ik}^{(1)} = \begin{cases} c(i, j), & (i, j) \in \mathbf{S}; \\ M, & (i, j) \notin \mathbf{S}, \end{cases}$$

де $\mathbf{S}$ – множина ланок транспортної мережі; M – число, що значно перевищує довжину будь-якого з маршрутів.

Далі формуються матриці $\mathbf{R}^{(2)}, \mathbf{R}^{(3)}, ..., \mathbf{R}^{(t+l)}, ..., \mathbf{R}^{(N)}$ з елементами

$$r_{ij}^{(t+l)} = \min\left\{r_{ij}^{(t)},\ r_{ij}^{(l)},\ \min_k\left(r_{ik}^{(t)} + r_{kj}^{(l)}\right)\right\},$$

де $t, l = 1, 2, ..., N-1$; $t + l = 2, 3, ..., N$.

Процес формування матриць припиняється, коли нова матриця збігається з попередньою, тобто $\mathbf{R}^{(N)} = \mathbf{R}^{(N-1)}$.

Закінчується етап перетворенням матриці відстаней $\mathbf{D}$ у зведену матрицю $\mathbf{D}''$.

Матриця $\mathbf{D}''$ формується за формулами методу «гілок і границь». Так, елементи матриці $\mathbf{D}''$, відповідно до методу «гілок і границь» визначаються виразом

$$d''_{ij} = d'_{ij} - \min_i d'_{ij}\ ,\ i, j = \overline{1, N}, \tag{2.6}$$

де

$$d'_{ij} = d_{ij} - \min_{j} d_{ij}, \quad i, j = \overline{1, N}. \qquad (2.7)$$

Зміст елементів матриці $\mathbf{D}''$ такий: пари пунктів, яким у цій матриці відповідають нулі, відповідно до методу «гілок і границь», рекомендується включити в кільцевий маршрут, що проходить через усі вершини транспортної мережі [5].

Очевидно, що розбиття мережі на райони буде ефективним, якщо в один район включати пари пунктів, яким відповідають найменші числа в зведеній матриці $\mathbf{D}''$.

Другий етап. На другому етапі починається формування першого району транспортної мережі. Перший пункт цього району має номер $j_{1,1}$. Розбиття цілком залежить від цього пункту. Процедура вибору пункту $j_{1,1}$ починається з обчислення характеристики середньої віддаленості кожного пункту від інших пунктів транспортної мережі за формулою

$$L_{ср.}(j) = \frac{1}{2(N-1)} \left(\sum_{\substack{i=1 \\ i \neq j}}^{N} d_{ij} + \sum_{\substack{i=1 \\ i \neq j}}^{N} d_{ji} \right), \quad j = \overline{1, N}. \qquad (2.8)$$

Для одержання найкращого результату за допомогою цього алгоритму треба було би розглянути *N* можливих розбивок *Розб.*–1, *Розб.*–2, ..., *Розб.*–*N*, вибираючи за пункт $j_{1,1}$ по черзі всі пункти транспортної мережі: $j_{1,1-P.1} = 1$, $j_{1,1-P.2} = 2$, ... Проте практичні дослідження показують, що частіше усього оптимальним виявляється розбивка, для якої в пункті $j_{1,1}$ оцінка $L_{ср.}(j_{1,1})$ є найбільшою. У залежності від вимірності транспортної мережі та обчислювальних можливостей комп'ютера, на якому виконується програма розбиття, пропонується розглядати тільки K із N розбивок. Причому $K = \dfrac{N}{2}$ для транспортних мереж із загальною кількістю пунктів $N \le 50$, $K = \dfrac{N}{3}$ для транспортних мереж, для яких $50 < N \le 100$, і $K = 5 \div 15$ при $N > 100$.

Першій пункт першого району розбиття повинний відповідати умові

$$L_{ср.}\left(j_{1,1-Р.1}\right) > L_{ср.}\left(j_{1,1-Р.2}\right) > L_{ср.}\left(j_{1,1-Р.3}\right) > \ldots$$

Тобто для першого варіанта розбиття перший пункт $j_{1,1}$ вибирається, виходячи з умови

$$L_{ср.}\left(j_{1,1}\right) = \max_{j \in \mathbf{J}} \left\{ L_{ср.}(j) \right\}, \tag{2.9}$$

де $\mathbf{J}$ – множина пунктів, що ще не були розподілені між районами, $\mathbf{J} = \{1,\ 2,\ \ldots,\ N\}$. А для розбивки k, $k \in \{2,3,\ldots,K\}$, виходячи з умови

$$L_{ср.}\left(j_{1,1}\right) = \max_{j \in \mathbf{J}/\{j_1, j_2, \ldots, j_{k-1}\}} \left\{ L_{ср.}(j) \right\}, \tag{2.10}$$

де j_1 збігається з номером $j_{1,1}$, що був вибраним при формуванні розбивки *Розб.*–1, $L_{ср.}\left(j_{1,1}\right) = \max\limits_{j \in \mathbf{J}} \left\{ L_{ср.}(j) \right\}$; j_2 – з номером пункту $j_{1,1}$, що був вибраним при формуванні розбивки *Розб.*–2, $L_{ср.}\left(j_{1,1} = j_2\right) = \max\limits_{j \in \mathbf{J}/\{j_1\}} \left\{ L_{ср.}(j) \right\}$, і т.д.

Потім до першого району приєднуються пункти $j_{2,1}$, $j_{3,1}$, ..., $j_{M,1}$. Порядок приєднання інших пунктів від вибору першого пункту $j_{1,1}$ не залежить. Кількість пунктів, що приєднуються на цьому етапі, визначається виразом

$$M = \left[q\frac{N}{R} \right].$$

Тут коефіцієнт q підбирається експериментально. При розв'язанні задач малої вимірності, до 20 пунктів, кращі результати маємо при $q = \frac{1}{2}$. Для задач великої вимірності більш прийнятним є $q = \frac{1}{3}$.

Приєднання до району, що формується, пункту $j_{s,1}$, $s \in \left\{\overline{2,M}\right\}$, здійснюється за критерієм $E_{j_{s,1}}$

$$E_{j_{s,1}} = \min_{\substack{j=\overline{1,N} \\ j \neq j_{t,1},\ t=\overline{1,s-1}}} \left\{E_j\right\}, \tag{2.11}$$

де E_j обчислюється відповідно до формули

$$E_j = \min\left\{ \min_{i \in \left\{j_{1,1}, j_{2,1}, \ldots, j_{s-1,1}\right\}} d''_{ij}, \quad \min_{i \in \left\{j_{1,1}, j_{2,1}, \ldots, j_{s-1,1}\right\}} d''_{ji} \right\}. \tag{2.12}$$

Третій етап. На черговому етапі починається формування інших районів. Для r-го району, $r \in \left\{\overline{2,R}\right\}$, номер початкового пункту $j_{1,r}$ визначається за критерієм

$$L'_{cp.}\left(j_{1,r}\right)=\max_{j\in\mathbf{J}}\left\{L'_{cp.}\left(j\right)\right\},\qquad(2.13)$$

де характеристики $L'_{cp.}\left(j\right)$ обчислюються за формулою

$$L'_{cp.}\left(j\right)=\frac{1}{2M\left(r-1\right)}\left(\sum_{\substack{i\in\mathbf{I}_t\\t=1,\ldots,r-1}}d_{ij}+\sum_{\substack{i\in\mathbf{I}_t\\t=1,\ldots,r-1}}d_{ji}\right),\quad j\in\mathbf{J}.\qquad(2.14)$$

До кожного з районів приєднується $\left(M-1\right)$ пункт. Критерій приєднання пункту $j_{s,r}$, $s\in\left\{2,3,\ldots,M\right\}$, є аналогічним критерію, що застосовувався на попередньому етапі. Пункт $j_{s,r}$ вибирається за критерієм

$$E_{j_{s,r}}=\min_{j\in\mathbf{J}}\left\{E_j\right\},\qquad(2.15)$$

де

$$E_j=\min\left\{\min_{i\in\mathbf{I}_r}d''_{ij},\ \min_{i\in\mathbf{I}_r}d''_{ji}\right\},\qquad(2.16)$$

I_r – множина пунктів, що приєднані до району *r*.

Таким чином, одержуємо *R* районів, кожний із яких містить у собі *M* пунктів. Крім того, є $\left(N-MR\right)$ пунктів, що не були віднесені до жодного із районів.

Четвертий етап. На останньому етапі здійснюється розподіл $\left(N-MR\right)$ пунктів, що залишилися нерозподіленими на третьому етапі.

Для кожного району визначають довжину кільцевого маршруту, що проходить через усі пункти. Обчислюють середню довжину отриманих маршрутів і накладають заборону на додавання нових пунктів до тих районів, «розмір» яких перевищує середнє значення більш ніж на p відсотків. Величина p задається в залежності від припустимої різниці між розмірами районів.

На практиці одержати райони з однаковими розмірами, тобто з однаковою довжиною маршруту комівояжера, або з розмірами, що розрізняються на задану кількість відсотків, можливо далеко не завжди. Причина цього – дискретність розмірів районів. Крім того, оцінка довжини маршруту комівояжера є досить трудомісткою задачею для районів із великою кількістю пунктів. Тому ретельний «контроль» розмірів районів може привести до надмірної кількості необхідних обчислень. При цьому суттєво скоротиться вимірність транспортних мереж, що можна розбити на райони за цим алгоритмом.

Таким чином, на практиці параметр p буде впливати на різницю між розмірами районів, але фактична різниця може від нього відрізнятися.

Для чергового району, на додавання пунктів до якого не була введена заборона, визначаємо найбільш вигідні для приєднання пункти. Нехай у поточному районі r уже міститься s пунктів. Тоді, у випадку приєднання чергового пункту до району кількість пунктів у районі стане рівною $s+1$. Для визначення номера цього пункту, а саме пункту $j_{s+1,r}$, використовується критерій

$$E'_{j_{s+1,r}} = \min_{j \in \mathbf{J}} \left\{ E'_j \right\}, \qquad (2.17)$$

де

$$E'_j = \min\left\{\begin{array}{l} \min\left\{\begin{array}{l} d''_{i1j} + d''_{i2j} \middle| d''_{i1j}, d''_{i2j} \le \\ \qquad \le d''_{ij},\ i, i1, i2 \in \mathbf{I}_r,\ i \ne i1,\ i \ne i2 \end{array}\right\}, \\ \min\left\{\begin{array}{l} d''_{ji1} + d''_{ji2} \middle| d''_{ji1}, d''_{ji2} \le \\ \qquad \le d''_{ji},\ i, i1, i2 \in \mathbf{I}_r,\ i \ne i1,\ i \ne i2 \end{array}\right\} \end{array}\right\}, \tag{2.18}$$

якщо $s+1 \in \{[qN/R]+1,\quad [2qN/R]\}$;

$$E'_j = \min\left\{\begin{array}{l} \min\left\{\begin{array}{l} d''_{i1j} + d''_{i2j} + d''_{i3j} \middle| d''_{i1j}, d''_{i2j}, d''_{i3j} \le d''_{ij}, \\ i, i1, i2, i3 \in \mathbf{I}_r, i \ne i1, i \ne i2, i \ne i3 \end{array}\right\}, \\ \min\left\{\begin{array}{l} d''_{ji1} + d''_{ji2} + d''_{ji3} \middle| d''_{ji1}, d''_{ji2}, d''_{ji3} \le d''_{ji}, \\ i, i1, i2, i3 \in \mathbf{I}_r, i \ne i1, i \ne i2, i \ne i3 \end{array}\right\} \end{array}\right\}, \tag{2.19}$$

якщо $s+1 \in \{[2qN/R]+1, [N/R]\}$.

Нехай для приєднання до району r згідно зі зазначеним критерієм (2.17) був відібраний пункт j_m ($j_m = j_{s+1,r}$). Для цього пункту визначається «найближчий» район. «Найближчим» будемо вважати район r_k, для якого справедливе співвідношення:

$$E''_{r_k, j_m} = \min_{r_t \in \{1, \dots R\}} \left\{E''_{r_t, j_m}\right\}, \tag{2.20}$$

$$E''_{r_t,j_m} = \min\left\{\begin{array}{l} \min\left\{\begin{array}{l} d''_{i1j_m} + d''_{i2j_m} \left| d''_{i1j_m}, d''_{i2j_m} \le d''_{ij_m}, \right. \\ i \in \mathbf{I}_r,\ i \ne i1, i \ne i2 \end{array}\right\} \\ \min\left\{\begin{array}{l} d''_{j_m i1} + d''_{j_m i2} \left| d''_{j_m i1}, d''_{j_m i2} \le d''_{j_m i}, \right. \\ i \in \mathbf{I}_r,\ i \ne i1,\ i \ne i2 \end{array}\right\} \end{array}\right\}. \quad (2.21)$$

Якщо «найближчим» районом r_k є район *r,* то пункт j_m приєднується до нього.

Потім пошук «найближчого пункту» повторюється для наступного району, до якого можна приєднувати пункти. Знову виконуємо перевірку, чи є район «найближчим» для вибраного пункту, і т.д.

Процес завершується, коли множина **J**, що складається із не приєднаних пунктів, стає порожньою.

2.3.1. Алгоритм модифікованого методу «гілок і границь»

Нижче подається алгоритм запропонованої модифікації методу «гілок і границь», що дозволяє більш точно визначити порядок дій.

Крок 1. Робиться введення відстаней між пунктами транспортної мережі (тільки для тих пар пунктів, що мають безпосередній транспортний зв'язок між собою, тобто без будь-яких транзитних пунктів).

Крок 2. Формується матриця відстаней між пунктами **D**. Відстань у матриці **D** – це довжина шляху між пунктами, або довжина найкоротшого маршруту, що з'єднує пункти.

Крок 3. Для всіх пунктів $j = \overline{1, N}$ обчислюються оцінки середньої відстані до пунктів $L_{ср.}(j)$ за формулою (2.8).

Крок 4. Обчислюється зведена матриця $\mathbf{D}''$ за формулами (2.6), (2.7).

Крок 5. Номер варіанта розбивки, що буде отриманий, дорівнюється до одиниці: $k := 1$.

Крок 6. Номер району, що буде сформований, дорівнюється до одиниці, $r := 1$. Формується множина **J** пунктів, що ще не були приєднані до жодного із районів. На початку реалізації алгоритму до них відносяться всі пункти транспортної мережі, тобто $\mathbf{J} := \{1, 2, ..., N\}$. Задаються множини пунктів $\mathbf{I}_1$, $\mathbf{I}_2$, …, $\mathbf{I}_R$ відповідно в районах 1, 2, …, *R*. Оскільки до них ще не був приєднаний жодний пункт, то $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$.

Крок 7. Задається початкове значення лічильника пунктів, що приєднуються до району: $s := 1$.

Крок 8. Визначається перший пункт першого району $j_{1,1}$, виходячи з умови (2.9), якщо $k = 1$, або з умови (2.10), якщо $k > 1$. Вибраний пункт $j_{1,1}$ переміщується з множини **J** у множину $\mathbf{I}_1$. З цього моменту кількість пунктів у множині $\mathbf{I}_1$ стає рівною одиниці, тобто тепер заданий номер пункту *s*, що приєднується до району, можна розглядати також як потужність множини пунктів району: $s = card\, \mathbf{I}_1$.

Крок 9. Номер поточного пункту району *s* збільшується на одиницю: $s := s + 1$.

Крок 10. За критерієм (2.11) – (2.12) здійснюється вибір пункту $j_{s,1}$ для приєднання його до першого району. Вибраний пункт $j_{s,1}$ переміщується з множини $\mathbf{J}$ в множину $\mathbf{I}_1$.

Крок 11. Якщо кількість пунктів у першому районі $card\,\mathbf{I}_1 = s < M$, то відбувається повернення до кроку 9.

Крок 12. Номер поточного району збільшується на одиницю: $r := r + 1$.

Крок 13. Номеру пункту, що приєднується до району, надається значення один: $s := 1$.

Крок 14. Вибирається перший пункт для приєднання до поточного району $j_{1,r}$ згідно з критерієм (2.13) – (2.14). Вибраний пункт $j_{1,r}$ переміщується з множини $\mathbf{J}$ в множину $\mathbf{I}_r$.

Крок 15. Номер поточного пункту в районі збільшується на одиницю: $s := s + 1$.

Крок 16. Вибирається пункт $j_{s,r}$ згідно з критерієм (2.15) – (2.16). Пункт $j_{s,r}$ переміщується з множини $\mathbf{J}$ в множину $\mathbf{I}_r$.

Крок 17. Якщо кількість пунктів у поточному районі $card\,\mathbf{I}_r = s < M$, то здійснюється повернення до кроку 15.

Крок 18. Номер поточного району *r* дорівнюється до одиниці: $r := 1$.

Крок 19. Визначається «розмір» кожного з районів $Size(r),\ r = \overline{1,R}$, тобто довжина маршруту комівояжера для кожного з отриманих районів, а також середнє арифметичне $\overline{Size}$ розмірів всіх отриманих районів за формулою

$$\overline{Size} = \frac{1}{R} \sum_{r=1}^{R} Size(r) \ .$$

Такі обчислення робляться тільки за умови, що з часу попереднього обчислення «розмірів» районів відбувалися зміни в складах множин $\mathbf{J}$ і $\mathbf{I}_r$, $r = \overline{1,R}$. Щоб прискорити швидкодію алгоритму рекомендується визначати «розміри» районів після приєднання декількох пунктів до районів.

Крок 20. Для поточного району перевіряється виконання умови

$$Size(r) \leq \overline{Size} \cdot (p+1) \ .$$

У випадку, якщо ця умова не виконується, відбувається перехід до кроку 25.

Крок 21. Здійснюється вибір пункту $j_{s+1,r}$ згідно з критерієм (2.17), що обчислюється з використанням формули (2.18) або (2.19) у залежності від величини $(s+1)$. Тут s – кількість пунктів у поточному районі, $s = card\, I_r$.

Крок 22. Для пункту $j_{s+1,r}$ здійснюється вибір «найближчого» району r_k за критерієм (2.20) – (2.21).

Крок 23. Якщо номер «найближчого» до пункту $j_{s+1,r}$ району r_k і номер поточного району r не збігаються, тобто $r_k \neq r$, то відбувається перехід до кроку 25.

Крок 24. Пункт $j_{s+1,r}$ переміщується з множини $\mathbf{J}$ в множину $\mathbf{I}_r$.

Крок 25. Якщо номер поточного району $r < R$, то реалізується наступний крок. У протилежному випадку відбувається перехід до кроку 27.

Крок 26. Номер поточного району збільшується на одиницю: $r := r + 1$. Здійснюється перехід до кроку 28.

Крок 27. Номеру поточного району надається значення один: $r := 1$.

Крок 28. Якщо множина $\mathbf{J}$ не порожня, тобто $\mathbf{J} \neq \varnothing$, то здійснюється повернення до кроку 19.

Крок 29. Обчислюється оцінка отриманої розбивки $Eval._{cur.}$, тобто визначається сумарна довжина маршрутів комівояжера за всіма районами: $Eval._{cur.} = \sum_{r=1}^{R} Size(r)$.

Крок 30. Якщо був знайдений тільки перший варіант розбиття, тобто $k = 1$, то відбувається перехід до кроку 32.

Крок 31. Якщо оцінка отриманої розбивки $Eval._{cur.}$ більше або ж збігається з найкращою з отриманих оцінок $Eval._{\min.}$, тобто $Eval._{cur.} \geq Eval._{\min}$, то відбувається перехід до кроку 33.

Крок 32. Оцінка отриманої розбивки зберігається як найкраща з оцінок: $Eval._{\min.} := Eval._{cur.}$. Отриманий розподіл пунктів за районами зберігається як найкраща розбивка.

Крок 33. Номер варіанта розбивки збільшується на одиницю: $k := k + 1$.

Крок 34. Якщо ще не перевищена задана кількість розбивок, тобто $k \le K$, то відбувається повернення до кроку 6.

Крок 35. На екран дисплея виводиться розподіл пунктів за районами, що відповідає найкращій з розбивок, тобто множини $\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_R$, і оцінка найкращої розбивки $Eval._{\min}$.

Послідовність дій при розбитті транспортної мережі на райони за допомогою модифікованого методу «гілок і границь» можна представити у вигляді блок-схеми, поданої на рис. 2.1 – 2.2.

При розробці модифікованого методу «гілок і границь» до задачі районування на кількість пунктів у транспортній мережі накладалася вимога: загальна кількість пунктів N повинна бути кратною кількості районів, тобто

$$N = kR, \quad k \in \{2,3,...\}. \tag{2.22}$$

Вимога (2.22) гарантує можливість отримання розбивки з однаковою кількістю пунктів у всіх районах:

$$card\, \mathbf{I}_r = \frac{N}{R} = k, \quad r = \overline{1, R}.$$

Далі описані всі алгоритми прийнятних модифікацій методу «гілок і границь», що пройшли тестування в однакових умовах – для одних і тих самих транспортних мереж. Всі алгоритми розв’язання задачі районування транспортної мережі оцінювалася за допомогою порівняння результатів розбиття з оптимальним результатом, що був отриманий за допомогою повного перебору всіх можливих варіантів розбиття. Порівняння робилося тільки для варіантів розбиття, що відповідали умові (2.22).

Рис. 2.1 – Початок схеми алгоритму модифікованого методу «гілок і границь»

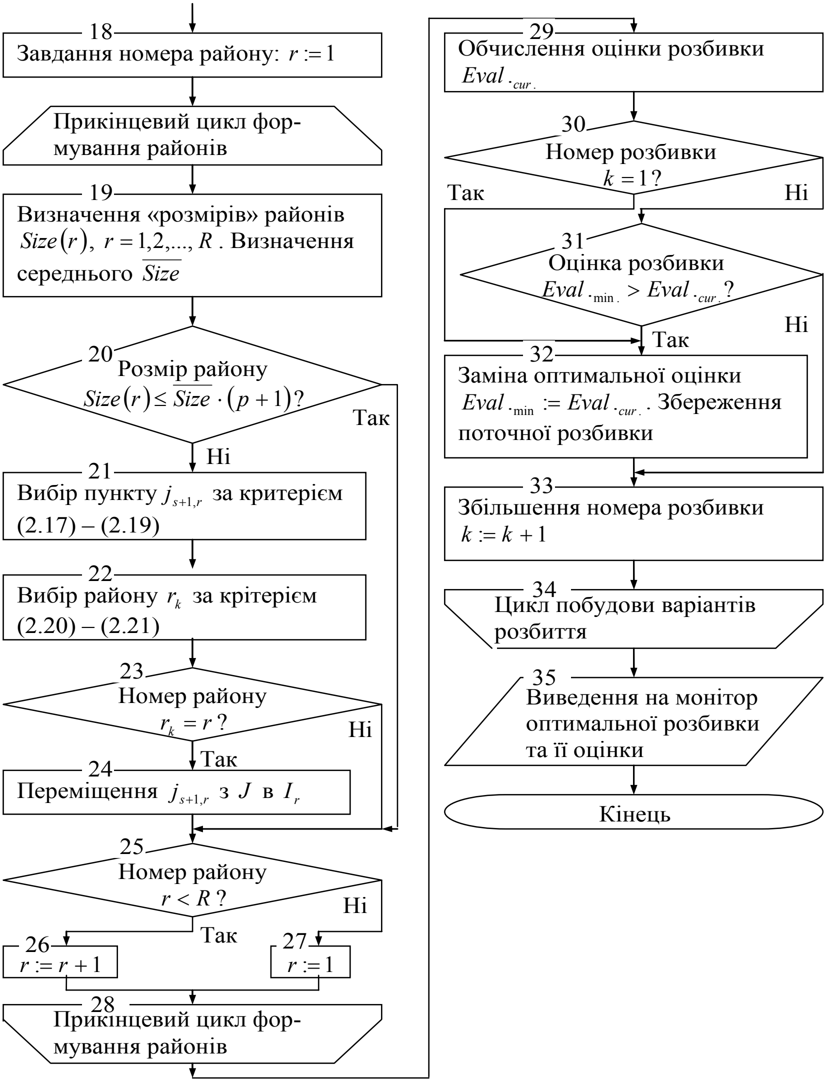


Рис. 2.2 – Закінчення схеми алгоритму модифікованого методу «гілок і границь»

2.3.2. Перший варіант алгоритму модифікованого методу «гілок і границь»

Крок 1. Уводиться матриця відстаней $\mathbf{D}$.

Крок 2. Відповідно до формул (2.6), (2.7) формується зведена матриця $\mathbf{D}''$ і вектор $\widetilde{\mathbf{d}}$, елементи якого визначаються за формулою

$$\widetilde{d}_j = \frac{\sum\limits_{\substack{i=1 \\ i \neq j}}^{N} d_{ij}}{N}, \quad j = \overline{1, N} . \tag{2.23}$$

Крок 3. Установлюється номер поточного району: $r := 1$, формуються множина пунктів, що не приєднані до жодного з районів, тобто $\mathbf{J} = \{1,2,...,N\}$, і порожні множини пунктів, що належать до районів розбиття: $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$.

Крок 4. Завдається початковий номер пункту поточного району: $s := 1$.

Крок 5. Вибирається перший елемент $j_{1,r}$ району r згідно з критерієм

$$\widetilde{d}_{j_{1,r}} = \max_{j \in \mathbf{J}} \widetilde{d}_j \tag{2.24}$$

і переміщується з множини $\mathbf{J}$ у множину $\mathbf{I}_r$.

Крок 6. Збільшується на одиницю кількість елементів у районі: $s := s + 1$.

Крок 7. Вибирається елемент $j_{s,r}$ за критерієм (2.15), (2.16), і відповідний пункт переміщується з множини $\mathbf{J}$ у множину $\mathbf{I}_r$.

Крок 8. Якщо кількість елементів у районі $s < \dfrac{N}{R}$, то здійснюється перехід до кроку 6.

Крок 9. Якщо $r < R$, то номер району збільшується на одиницю: $r := r + 1$, і здійснюється повернення до кроку 5.

Крок 10. Отриманий результат виводиться на екран монітору. Він являє собою списки пунктів, що треба віднести до кожного з районів. Також виводяться довжини маршрутів комівояжера для кожного із цих районів і сумарна довжина маршрутів комівояжера за всіма районами.

Блок-схему алгоритму 1 наведено на рис. 2.3.

2.3.3. Другий варіант алгоритму модифікованого методу «гілок і границь»

Крок 1. Уводиться матриця відстаней **D**.

Крок 2. Відповідно до формул (2.6), (2.7) формується зведена матриця $\mathbf{D}''$. Формується вектор $\widetilde{\mathbf{d}}$, елементи якого обчислюються за формулою (2.23).

Крок 3. Завдання множини пунктів, що не приєднані до жодного із районів: $\mathbf{J} = \{1, 2, ..., N\}$, порожніх множин пунктів, що належать до районів розбиття: $\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing$. Номер поточного пункту кожного із районів *s* дорівнюється одиниці: $s := 1$.

Крок 4. Вибираються перші елементи $j_{1,r}$ для всіх районів r, $r = \overline{1, R}$, згідно з критерієм (2.24). Вибрані елементи $j_{1,r}$ переміщуються з множини **J** у відповідні множини $\mathbf{I}_r$, $r = \overline{1, R}$.

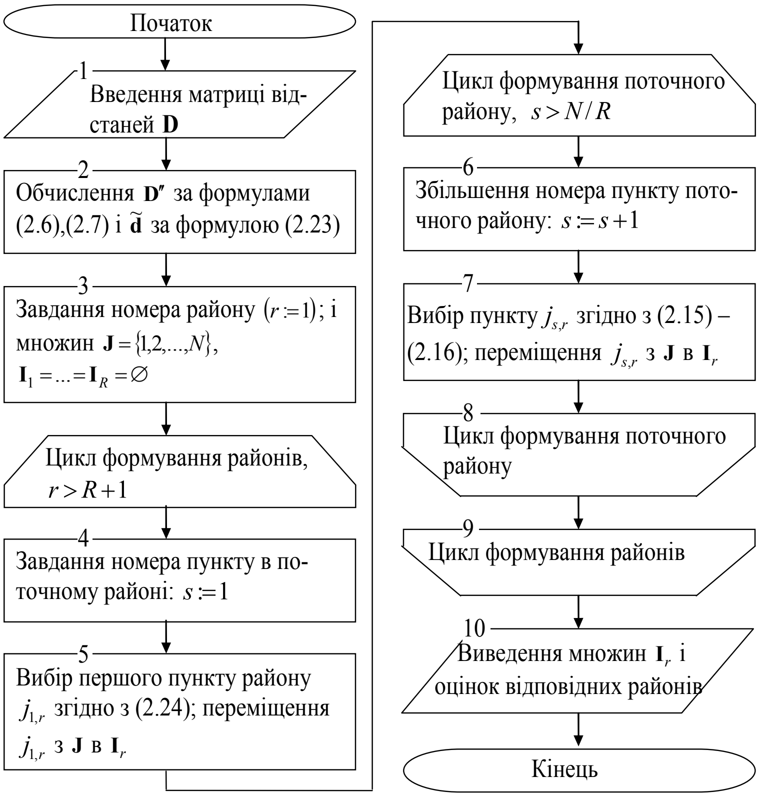


Рис. 2.3 − Схема алгоритму 1

Крок 5. Номер поточного пункту всіх районів збільшується на одиницю: $s:=s+1$.

Крок 6. Якщо кількість пунктів у районах $s=\dfrac{N}{R}+1$, то розбиття вважається завершеним і здійснюється перехід до кроку 11.

Крок 7. Номер поточного району дорівнюється одиниці: $r := 1$.

Крок 8. До множини пунктів поточного району $\mathbf{I}_r$ додається пункт $j_{s,r}$. Цей пункт $j_{s,r}$, $r = \overline{1,R}$ вибирається згідно критерію (2.15) – (2.16).

Крок 9. Номер поточного району збільшується на 1: $r := r + 1$.

Крок 10. Якщо номер поточного району перевищує R, тобто $r > R$, то здійснюється перехід до кроку 5. У противному випадку – перехід до кроку 8.

Крок 11. На екран виводяться списки пунктів, що рекомендується віднести до кожного з районів, а також довжини маршрутів комівояжера для кожного з цих районів і їх сумарна довжина.

Блок-схема алгоритму 2 зображена на рис. 2.4.

2.3.4. Третій варіант алгоритму модифікованого методу «гілок і границь»

Крок 1. Уводиться матриця відстаней $\mathbf{D}$.

Крок 2. Відповідно до формул (2.1), (2.2) формується зведена матриця $\mathbf{D}''$. Формується вектор $\widetilde{\mathbf{d}}$, елементи якого обчислюються по формулі (2.23).

Крок 3. Установлюється: номер поточного району $(r := 1)$; множина пунктів $(\mathbf{J} = \{1, 2, ..., N\})$, що не приєднані до жодного району; множини пунктів, що належать до районів $(\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing)$.

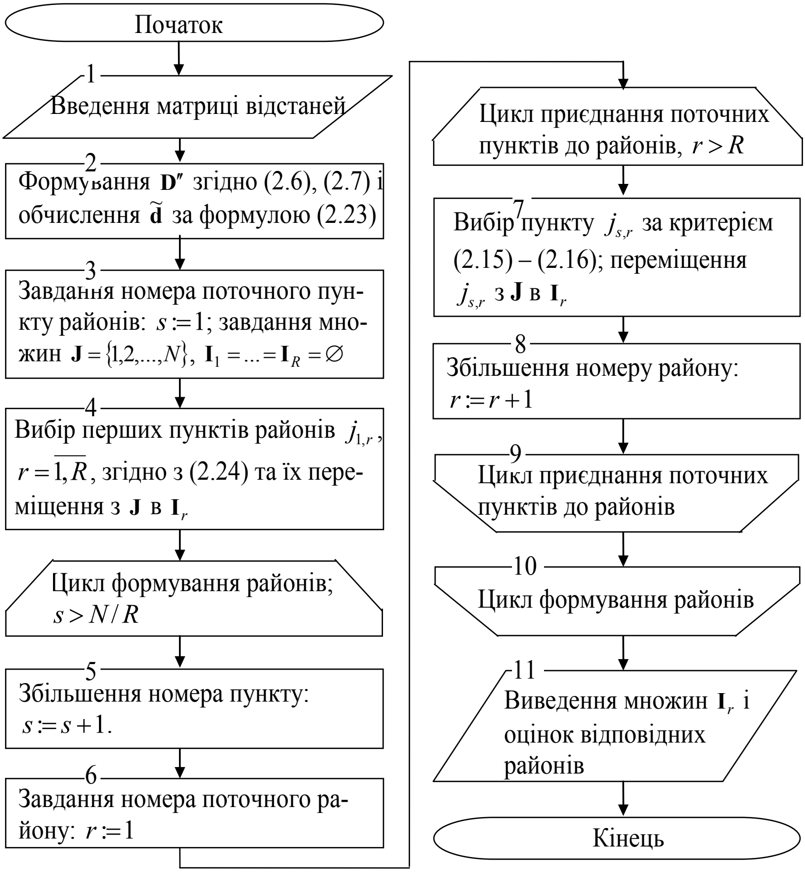


Рис. 2.4 − Схема алгоритму 2

Крок 4. Кількість елементів у поточному районі дорівнюється до одиниці, $s:=1$.

Крок 5. Вибирається перший елемент $j_{1,r}$ району r відповідно до критерію (2.24) і переміщується з множини $\mathbf{J}$ у множину $\mathbf{I}_r$.

Крок 6. Номер чергового елемента району збільшується на одиницю, $s := s + 1$.

Крок 7. Вибір пункту $j_{s,r}$ за критерієм

$$F_{j_{s,r}} = \min_{j \in \mathbf{J}} \{F_j\}, \tag{2.25}$$

де F_j для кожного пункту $j \in \mathbf{J}$ обчислюється як сума

$$F_j = \sum_{i \in \mathbf{I}_r} d''_{ij} \ . \tag{2.26}$$

Пункт $j_{s,r}$ переміщується з множини **J** у множину $\mathbf{I}_r$.

Крок 8. Якщо кількість елементів у районі $s < \frac{N}{R}$, то здійснюється перехід до кроку 6.

Крок 9. Номер району збільшується на одиницю: $r := r + 1$.

Крок 10. Якщо $r \le R$, то робиться перехід до кроку 5.

Крок 11. На екран виводяться списки пунктів доставки, що рекомендується віднести до кожного з районів, довжини маршрутів комівояжера для кожного району розбиття та їх сумарна довжина.

Блок-схема алгоритму 3 наведена на рис. 2.5.

2.3.5. Четвертий варіант алгоритму модифікованого методу «гілок і границь»

Крок 1. Уводиться матриця відстаней **D**.

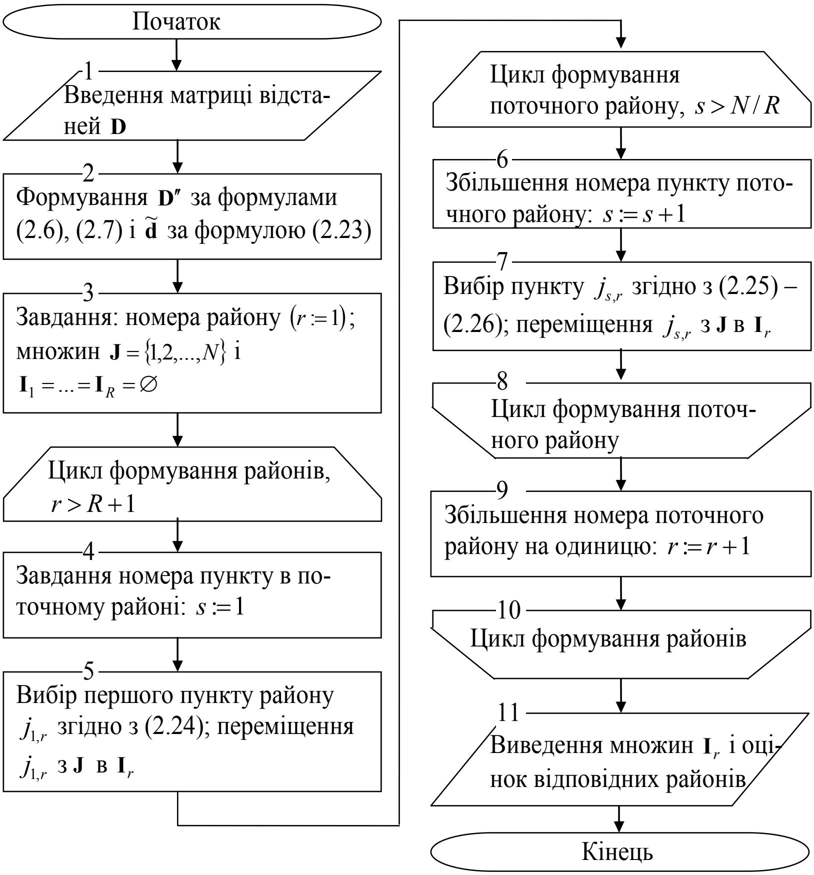


Рис. 2.5 − Схема алгоритму 3

Крок 2. Формується зведена матриця $\mathbf{D}''$ відповідно до формул (2.6), (2.7); вектор $\tilde{\mathbf{d}}$, елементи якого обчислюються за формулою (2.23).

Крок 3. Завдаються: номер поточного району $(r := 1)$; множина пунктів, не приєднаних до жодного із районів

$(\mathbf{J} = \{1, 2, ..., N\})$; множини пунктів, що належать до районів розбиття $(\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing)$.

Крок 4. Кількість елементів у поточному районі встановлюється рівною одиниці: $s := 1$.

Крок 5. Вибирається перший елемент $j_{1,r}$ району *r* згідно з критерієм (2.24). Елемент $j_{1,r}$ переміщується з множини $\mathbf{J}$ у множину $\mathbf{I}_r$.

Крок 6. Номер чергового елемента району збільшується на одиницю: $s := s + 1$.

Крок 7. Якщо номер поточного пункту $s \geq \frac{N}{2R}$, то здійснюється перехід до кроку 9.

Крок 8. Вибирається елемент $j_{s,r}$ згідно з критерієм (2.15) – (2.16). Здійснюється перехід до кроку 10.

Крок 9. Вибирається елемент $j_{s,r}$ згідно з критерієм (2.17), (2.18),

Крок 10. Пункт $j_{s,r}$ переміщується з множини **J** у множину $\mathbf{I}_r$.

Крок 11. Якщо кількість елементів у районі $s < \frac{N}{R}$, то здійснюється перехід до кроку 6.

Крок 12. Номер району збільшується на одиницю: $r := r + 1$.

Крок 13. Якщо $r \leq R$, то здійснюється повернення до кроку 4.

Крок 14. На екран монітору виводяться: списки пунктів доставки, що рекомендується віднести до кожного з районів; довжини маршрутів комівояжера для кожного отриманого району; сумарна довжина маршрутів комівояжера.

Блок-схема алгоритму 4 наведена на рис. 2.6.

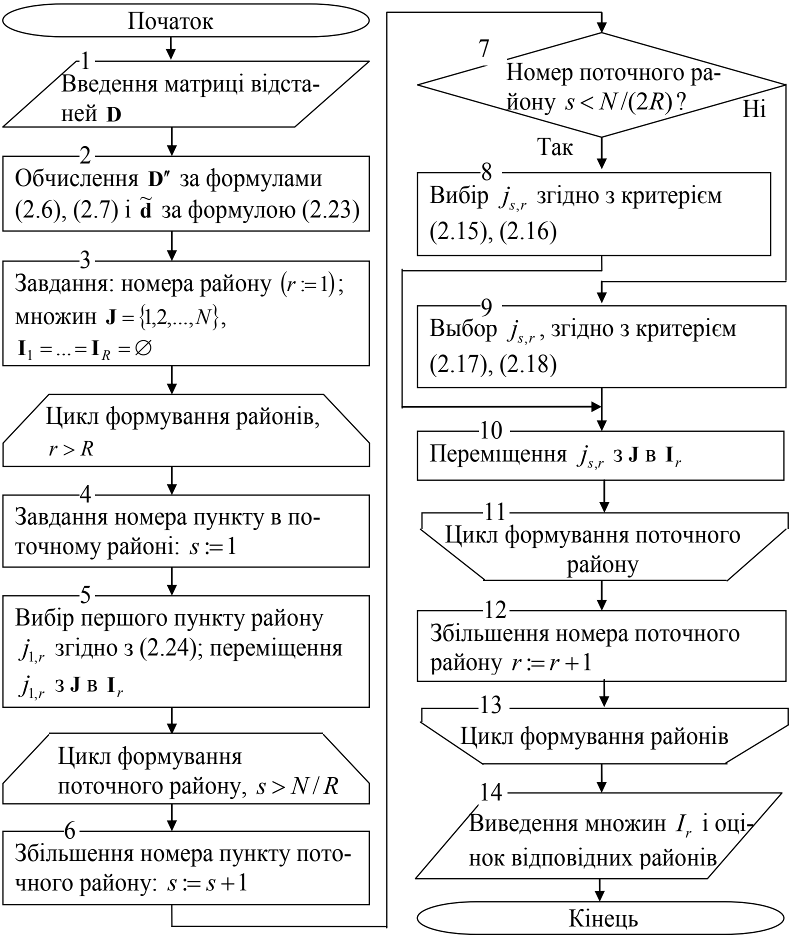


Рис. 2.6 − Схема алгоритму 4

2.3.6. П'ятий варіант алгоритму модифікованого методу «гілок і границь»

Крок 1. Уводиться матриця відстаней $\mathbf{D}$.

Крок 2. Формуються: зведена матриця $\mathbf{D}''$ відповідно до формул (2.6), (2.7); вектор $\tilde{\mathbf{d}}$, елементи якого обчислюються за формулою (2.23).

Крок 3. Завдаються: номер поточного району $(r := 1)$; множина пунктів, не приєднаних до жодного із районів $(\mathbf{J} = \{1, 2, ..., N\})$; множини пунктів, що належать до районів розбиття $(\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing)$.

Крок 4. Завдається номер пункту в поточному районі $(s := 1)$.

Крок 5. Вибирається перший елемент $j_{1,r}$ району r згідно з критерієм (2.24). Елемент $j_{1,r}$ переміщується з множини $\mathbf{J}$ у множину $\mathbf{I}_r$.

Крок 6. Номер чергового елемента району збільшується на одиницю $(s := s + 1)$.

Крок 7. Якщо номер поточного пункту $s < \dfrac{N}{3R}$, то здійснюється перехід до наступного кроку. Якщо $\dfrac{N}{3R} \le s < \dfrac{2N}{3R}$ – до кроку 9. Якщо $s \ge \dfrac{2N}{3R}$ – до кроку 10.

Крок 8. Вибирається елемент $j_{s,r}$ за критерієм (2.15), (2.16). Здійснюється перехід до кроку 11.

Крок 9. Вибирається елемент $j_{s,r}$ за критерієм (2.17), (2.18). Здійснюється перехід до кроку 11.

Крок 10. Вибирається елемент $j_{s,r}$ за критерієм (2.17), (2.19).

Крок 11. Пункт $j_{s,r}$ переміщується з множини $\mathbf{J}$ у множину $\mathbf{I}_r$.

Крок 12. Якщо кількість елементів у районі $s < \dfrac{N}{R}$, то здійснюється перехід до кроку 6.

Крок 13. Номер району збільшується на одиницю: $r := r + 1$.

Крок 14. Якщо номер району $r < R + 1$, то здійснюється повернення до кроку 4.

Крок 15. На екран монітору виводяться: списки пунктів доставки, що рекомендується віднести до кожного з районів; довжини маршрутів комівояжера для кожного отриманого району; сумарна довжина маршрутів комівояжера.

Блок-схема алгоритму 5 наведена на рис. 2.7.

2.3.7. Шостий варіант алгоритму модифікованого методу «гілок і границь»

Крок 1. Вводиться матриця відстаней $\mathbf{D}$.

Крок 2. Формується зведена матриця $\mathbf{D}''$ відповідно до формул (2.6), (2.7). За формулою (2.8) обчислюються оцінки $L_{ср.}(j)$ для всіх пунктів, $j = \overline{1, N}$.

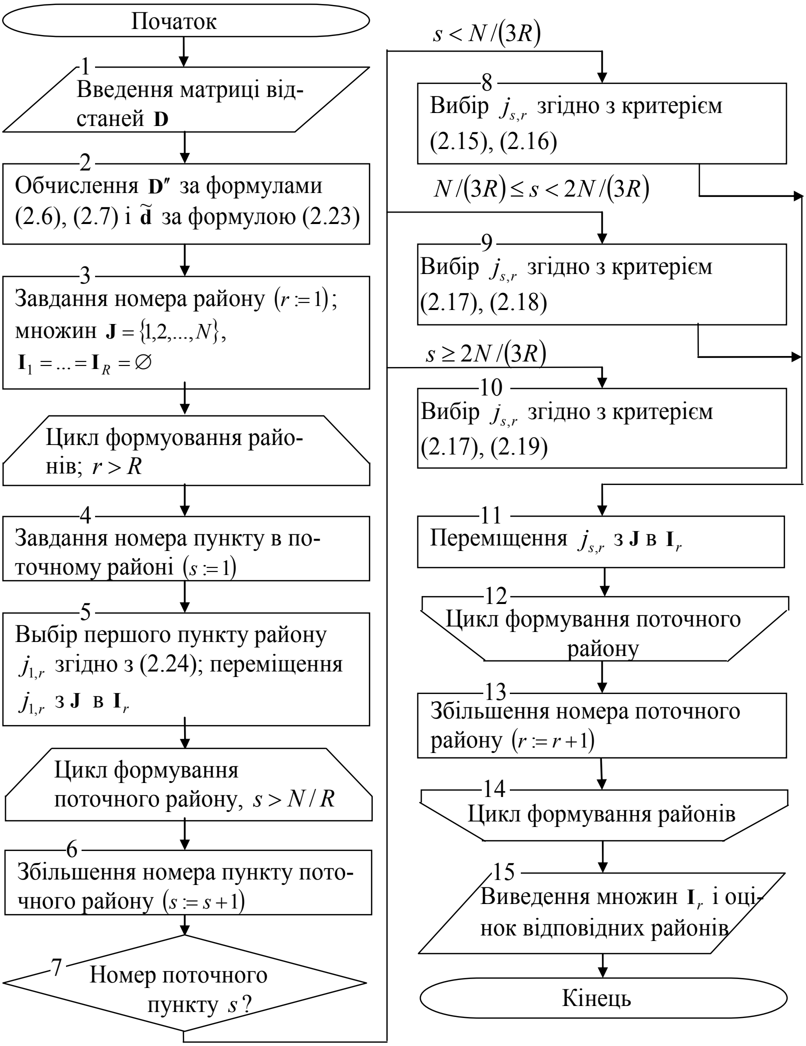


Рис. 2.7 − Схема алгоритму 5

Крок 3. Завдається: номер поточного району $(r := 1)$; множина пунктів, що не приєднані до жодного із районів $(\mathbf{J} = \{1, 2, ..., N\})$, множини пунктів, що належать до районів $(\mathbf{I}_1 = \mathbf{I}_2 = ... = \mathbf{I}_R = \varnothing)$; параметр *p*, що визначає відносну кількість пунктів, на яку число пунктів в районі не повинно перевищувати величину $\left\lceil \frac{N}{R} \right\rceil$.

Крок 4. Завдається номер елемента в поточному районі: $s := 1$.

Крок 5. Якщо номер поточного району $r = 1$, то здійснюється перехід до наступного кроку. У протилежному випадку – до кроку 7.

Крок 6. Вибирається перший елемент $j_{1,r}$ району *r* за критерієм (2.9). Здійснюється перехід до кроку 8.

Крок 7. Вибирається перший елемент $j_{1,r}$ району *r* за критерієм (2.13).

Крок 8. Елемент $j_{1,r}$ переміщається з множини $\mathbf{J}$ у множину $\mathbf{I}_r$.

Крок 9. Номер чергового елемента району збільшується на одиницю: $s := s + 1$.

Крок 10. Вибирається елемент $j_{s,r}$ за критерієм (2.15), (2.16).

Крок 11. Вибраний пункт $j_{s,r}$ переміщується з множини $\mathbf{J}$ у множину $\mathbf{I}_r$.

Крок 12. Якщо кількість елементів у районі $s < \dfrac{N}{2R}$, то здійснюється перехід до кроку 9.

Крок 13. Номер району збільшується на одиницю: $r := r + 1$.

Крок 14. Якщо номер району $r \le R$, здійснюється повернення до кроку 4.

Крок 15. Виконується присвоєння номера району: $r := 1$.

Крок 16. Якщо кількість пунктів у поточному районі не перевищує $\left[\dfrac{N}{R}\right] + Np$, то відбувається перехід до наступного кроку. У протилежному випадку здійснюється перехід до кроку 21.

Крок 17. Вибирається елемент $j_{s+1,r}$ за критерієм (2.17), (2.18).

Крок 18. Для вибраного елемента $j_{s,r}$ вибирається найближчий район r_k за критерієм (2.20), (2.21).

Крок 19. Якщо $r_k = r$, то здійснюється перехід до наступного кроку. У протилежному випадку – до кроку 21.

Крок 20. Кількість пунктів у розглянутому районі r збільшується на одиницю: $s := s + 1$. Вибраний пункт переміщується з множини $\mathbf{J}$ у множину $\mathbf{I}_r$.

Крок 21. Якщо $r < R$, то відбувається перехід до наступного кроку. У протилежному випадку – до кроку 23.

Крок 22. Номер району збільшується на одиницю: $r := r + 1$. Здійснюється перехід до кроку 24.

Крок 23. Виконується присвоєння $r := 1$.

Крок 24. Якщо множина $\mathbf{J}$ не порожня $(\mathbf{J} \neq \varnothing)$, то здійснюється повернення до кроку 16.

Крок 25. На екран монітору виводяться: списки пунктів доставки, що рекомендується віднести до кожного з районів; довжини маршрутів комівояжера для кожного отриманого району; сумарна довжина маршрутів комівояжера.

Блок-схема алгоритму 6 наведена на рис. 2.8.

2.4. Аналіз варіантів алгоритму модифікованого методу «гілок і границь»

У таблиці 2.1 наведені результати тестування розглянутих варіантів алгоритму модифікованого методу «гілок і границь» стосовно задачі розбиття транспортних мереж малої вимірності на райони.

З аналізу результатів тестування випливає, що найменше в середньому відхилення від оптимального результату дає алгоритм 4. Але при цьому має місце очевидна тенденція до збільшення відхилення від оптимального результату зі збільшенням кількості пунктів у транспортній мережі. В алгоритмі 3 (другого за малістю середнього значення) ця тенденція виявляється ще більш яскраво. В алгоритмі 6 (наступного за малістю середнього значення) відхилення від оптимального значення зростає найменше. Через те, що найбільш актуальним є розв'язання задач саме великої вимірності, остаточним варіантом адаптації методу «гілок і границь» для задачі розбиття транспортних мереж на райони вибраний алгоритм 6.

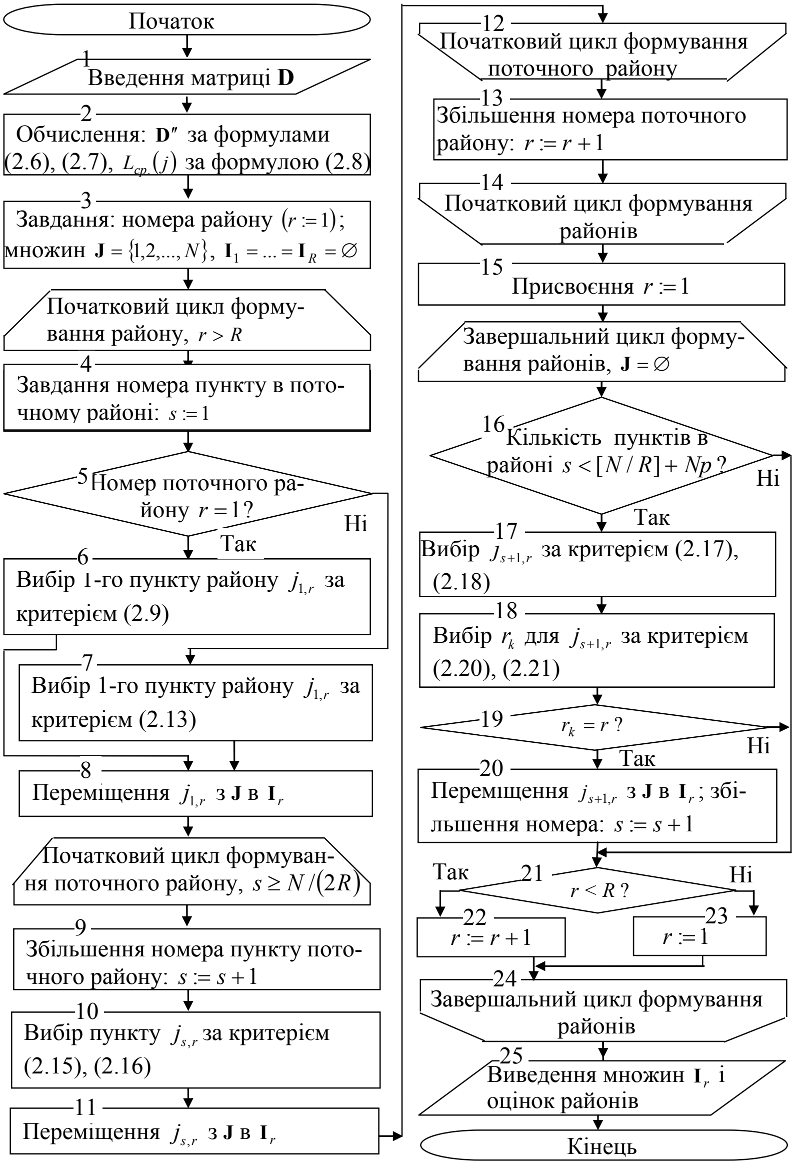


Рис. 2.8 – Схема алгоритму 6

Таблиця 2.1 – Результати тестування альтернативних алгоритмів модифікованого методу «гілок і границь» на транспортних мережах малої вимірності

Кількість пунктів в мережі	Відхилення від оптимального результату, %					
	Алгоритм 1	Алгоритм 2	Алгоритм 3	Алгоритм 4	Алгоритм 5	Алгоритм 6
Розбиття транспортної мережі на 2 райони						
8	1,2	14,8	0,7	1,6	0,7	0,7
10	3,6	13	0	1,6	1,6	1,9
12	10,5	23,5	5	5	5	5
14	6,9	14,4	4,5	2,3	8,4	3,3
16	10	26,3	5,9	6,5	8,2	5,1
18	0	17,7	10	5,1	8,2	0
Розбиття транспортної мережі на 3 райони						
9	5,7	11,8	1,8	0	8,2	12
12	3,7	24,5	0,4	0	1,1	6
15	0,6	9,7	0,6	0	0	0,6
	Середнє відхилення, %					
	4,7	17,3	3,2	2,4	4,6	3,8

РОЗДІЛ 3

РОЗВ’ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА ВЕЛИКОЇ ВИМІРНОСТІ ЗА ПРИНЦИПАМИ ДИНАМІЧНОГО ПРОГРАМУВАННЯ

На цей час для доставки вантажів широко застосовується автомобільний транспорт. У зв'язку з цим досить актуальною є задача організації доставки дрібних вантажів. Така задача звичайно зводиться до задачі комівояжера або задачі розвезення вантажу.

При розв’язання задачі комівояжера зазвичай прагнуть досягти мінімальної довжини маршруту, тобто мінімізувати довжину пробігу автотранспорту. Однак, використовуючи ті ж методи, можна легко перейти до мінімізації витрат часу, якщо враховувати не тільки довжини шляхів, що зв'язують пункти відправлення і доставки, але і швидкість, із якої можна рухатися на відповідних ділянках дороги, і час, що затрачається на зупинки в кожному пункті.

Задачу комівояжера можна також розглядати як допоміжну при розв’язанні задач, коли кінцевим результатом не є оптимальний маршрут комівояжера. Так, в попередньому розділі цієї монографії був запропонований алгоритм, при реалізації й оцінці якості якого необхідно виконувати побудову маршруту комівояжера та визначення його довжину досить багато разів.

Важливими критеріями роботи алгоритму для розв’язання задачі комівояжера є як точність результатів, так і швидкість їх одержання.

Існує досить багато методів, що забезпечують більш-менш точне розв’язання задачі комівояжера. Аналіз методів ускладнюється тим, що останні докладні дослідження з цієї теми виконувалися у восьмидесяті роки. У зв'язку з цим важко реально оцінити швидкодію всіх формальних (точних) і евристичних (наближених) методів, описаних у літературі [5]. У той же час виникає питання про точність евристичних методів, запропонованих у літературі, тобто про вплив вимірності задачі на точність результату розв’язання.

За загальним визнанням методи «гілок і границь» і динамічного програмування дозволяють одержувати точні результати. Проведене тестування показало, що метод динамічного програмування дозволяє вирішувати задачі більш високої вимірності в порівнянні з методом «гілок і границь». Але перевірка показала, що метод динамічного програмування не є точним. «Мінімальні» маршрути, отримані за його допомогою, можуть бути гірше маршрутів, отриманих за допомогою методу «гілок і границь».

3.1. Аналіз існуючих методів маршрутизації для перевезень дрібного вантажу

Як було відзначено вище, розв’язання задачі розвезення дрібного вантажу при певних умовах може бути зведене до розв’язання задачі комівояжера або задачі розвозу. Задача комівояжера припускає проходження всіх пунктів доставки на протязі одного рейсу, тобто будується тільки один маршрут. В задачі розвозу будується декілька маршрутів.

Методи розв’язання зазначених задач діляться на дві основні групи. Перша з них – точні методи, що гарантують об'єктивно оптимальні рішення. Друга – наближені методи, що дозволяють наблизитися до оптимального рішення з заданою точністю наближення.

Одним із відомих точних методів розв'язання задачі комівояжера є метод динамічного програмування [15]. Даний метод призначений для розв'язання широкого кола задач, що можуть бути розбиті на доповнюючи одна одну підзадачі, причому оптимальне рішення цих підзадач може бути використане для розв'язання вихідної задачі. Спочатку здійснюється розв'язання підзадач, а далі на підставі отриманих рішень знаходиться рішення вихідної задачі.

Стосовно до задачі комівояжера підзадачею є початкова частина маршруту, що містить в собі деяку кількість пунктів, що менша за загальну кількість. Рішення досягається за n ітерацій, де $(n+1)$ – загальна кількість пунктів.

Спочатку знаходяться найкоротші маршрути, що проходять через один пункт. Потім, на їх підставі – найкоротші маршрути, що проходять через 2 пункти, і так далі. На останній ітерації знаходяться маршрути, що проходять через усі пункти. Ітеративний процес завершується оптимальним замиканням маршруту.

При пошуку маршруту з пункту i у пункт 0 з k транзитними пунктами, $k=\overline{2,n}$, маршрут, що містить $k-1$ транзитний пункт, вибирається за критерієм

$$f(i; j_1, j_2, ..., j_k) = \\ = \min_{1 \le m \le k} \left\{ d_{ij_m} + f(j_m; j_1, j_2, ..., j_{m-1}, j_{m+1}, ..., j_k) \right\}. \quad (3.1)$$

де d_{ij_m} – довжина ланки транспортної мережі між пунктами i та j_m; $f(i; j_1, j_2, ..., j_k)$ – оцінка довжини маршруту з i в 0, що проходить через пункти $j_1, j_2, ..., j_k$.

При виборі маршруту з $(k-1)$-м транзитним пунктом не розглядаються маршрути, що містять пункт i, $i \notin \{j_1, j_2, ..., j_k\}$.

У випадку, коли на деякому кроці всі маршрути, що відповідають оцінкам $f(j_m; j_1, j_2, ..., j_{m-1}, j_{m+1}, ..., j_k)$, $j_m = \overline{1, n}$, містять пункт i, тобто не можуть бути використані для побудови, використовуються маршрути з $k-2$ транзитними пунктами, $k-3$ пунктами і так далі.

Час роботи такого алгоритму визначається еквіпотенціальною величиною $O\left(n^2 2^n\right)$. У зв'язку з особливостями алгоритму треба зберігати всі проміжні обчислення. Це приводить до того, що при реалізації алгоритму необхідний об'єм пам'яті ЕОМ також зростає в експоненціальній залежності від вимірності задачі [16]. Перевантаження пам'яті робить неможливим нормальне завершення процесу розв'язання задачі.

У [17] була запропонована модифікація методу динамічного програмування, що спрощує пошук оптимального рішення за допомогою корекції функцій стану. Корекція дозволяє заздалегідь зазначити варіанти, що перевищують верхню границю, отже, свідомо неоптимальні, і вилучити їх із розгляду.

Для вибракування елементів матриці пропонується використовувати відношення

$$p_{ij} = \frac{d_{i,j}}{\sum_{k=1}^{n} d_{i,k}}, \quad i = 1,2,...,n\ , \tag{3.2}$$

і залишати в матриці заздалегідь обумовлену кількість елементів, що мають найменше значення p_{ij}. Інші елементи матриці приймаються рівними нескінченно великому значенню. При цьому відхилення від оптимального результату складає 1 – 1,5 %. Така модифікація дозволяє знаходити квазіоптимальні рішення для задач вимірністю в 30 – 40 пунктів [17].

Серед точних методів найбільш широке визнання одержав метод «гілок і границь» [5,18]. Метод «гілок і границь» заснований на розбитті множини маршрутів об'їзду всіх пунктів транспортної мережі на підмножини з подальшим поступовим відсіюванням безперспективних підмножин. Процес продовжується до одержання підмножини, що містить один маршрут мінімальної довжини. Перспективними вважаються підмножини, що мають мінімальну оцінку, тобто довжини маршруту, що збігається довжиною оптимального маршруту комівояжера.

Графічно розбивка подається у вигляді дерева, вершинами якого є отримані підмножини. Для обчислення оцінок підмножин дані про транспортну мережу, що містять n пунктів, подаються у вигляді матриці $\mathbf{D} = \{d_{ij}\}_{n \times n}$, у якій занесені всі відстані між пунктами мережі.

Точна нижня границя маршруту комівояжера для всієї множини маршрутів L дорівнює сумі мінімальних елементів кожного з рядків матриці $\mathbf{D}$ та мінімальних елементів у стовпців матриці $\mathbf{D}$ після редукування по рядках:

$$L = \sum_{i=1}^{n} \min_{j \in \{1,\dots n\}} d_{ij} + \sum_{j=1}^{n} \min_{i \in \{1,\dots n\}} \left(d_{ij} - \min_{j \in \{1,\dots n\}} d_{ij} \right). \qquad (3.3)$$

Редукування – це зменшення всіх елементів рядка на величину його найменшого елемента. Редукування робиться на кожній ітерації для рядків і стовпців одержуваних матриць. Точна нижня границя підмножин оцінюється аналогічним способом, але при цьому використовуються відповідні їм матриці.

При розбивці вихідної множини та всіх проміжних підмножин щораз одержують дві непересічні підмножини. Одна з підмножин є множиною маршрутів, що містять деяку ланку (i, j) транспортної мережі, друга – множину маршрутів, що не містить цієї ланки. Точна нижня границя довжини маршруту комівояжера для першої із підмножин дорівнює точній нижній границі L вихідної множини, для другого з підмножин – сумі $L + f_{(i,j)}$, де $f_{(i,j)}$ – оцінка, що обчислена для ланки (i, j) за формулою

$$f_{(i,j)} = \min_{q=\underline{1,n}} \{d_{iq}\} + \min_{p=\underline{1,n}} \{d_{pj}\}. \tag{3.4}$$

Тут числа d_{iq} і d_{pj} – це елементи матриці відстаней після операцій зведення.

Для розбивки на підмножини вибирається ланка (i, j) з найбільшою оцінкою $f_{(i,j)}$:

$$f_{(i,j)} = \max_{p,q} \{f_{(p,q)}\} . \tag{3.5}$$

З розгляду виключаються ланки, що можуть викликати передчасне замикання (передчасний цикл) маршруту комівояжера.

У випадку, якщо на якомусь із етапів виявляється, що нижня границя якоїсь із підмножин, залишених не розділеними, менше, ніж нижня границі знов отриманих підмножин, то досягнутий результат зберігається, і розгалуження дерева продовжується з вершини, якій відповідає найменша нижня границя.

Метод може застосовуватися для розбиття транспортної мережі, що складається із 40 – 50 пунктів [5].

Метод «гілок і границь» може бути також адаптований для розв'язання задачі розвезення [5]. Для цього створюється m копій пункту відправлення вантажу, m відповідає необхідній кількості маршрутів. Після розв'язання задачі комівояжера для такої розширеної транспортної мережі копії сполучаються в один пункт. У результаті утворюється потрібна кількість маршрутів оптимальної довжини. При необхідності в процесі побудови маршрутів на кожному кроці аналізується загальний об'єм замовлення на вантаж у пунктах, що включені у маршрут. Маршрути, у яких об'єм замовлення перевищує вантажопідйомність використовуваного транспортного засобу, блокуються.

Іноді з метою економії часу, що затрачається на розв'язання, від пошуку оптимуму відмовляються.

У роботі [10] автор пропонує наступні шляхи для зменшення витрат часу та необхідного об'єму потрібної пам'яті ЕОМ:

- відмова від пошуку точного рішення;
- заміна створення копій пункту відправлення вантажу на дозвіл багатократної появи даного пункту в маршруті;
- блокування в зведеній матриці максимальних елементів, поява яких у маршруті є малоймовірною;
- динамічне агрегування пунктів доставки.

Ще один точний метод розв'язання задачі комівояжера запропонований А.В.Крушевським [19]. Суть методу полягає в одержанні такого порядку пунктів призначення, який відповідає порядку їх проходження в маршруті мінімальної довжини. Для одержання цього порядку застосовується ітеративний процес. На кожному кроці за даними матриці $\mathbf{D} = \{d_{ij}\}$, що відповідає поточному порядку пунктів, обчислюються допоміжні величини:

$$l(i, j) = d_{i-1j} + d_{ij+1} + d_{j-1i} + d_{ji+1} - d_{ii+1} - d_{i-1i} - d_{j-1j} - d_{jj+1},$$

$$j \neq i + 1\ , \qquad (3.6)$$

$$l(i, j) = d_{i-1i} + d_{ii+2} + d_{i+1i} - d_{i-1i} - d_{ii+1} - d_{i+1i+2},$$

$$j = i + 1, \quad i, j = \overline{2, n}\ . \qquad (3.7)$$

Величина $l(i, j)$ показує, наскільки збільшиться або зменшиться довжина маршруту при транспозиції елементів i та j.

Після обчислення величин (3.6), (3.7) відбирається множина пар елементів $\{(i_1, j_1), (i_2, j_2), \ldots, (i_s, j_s)\}$, яким відповідають мінімальні оцінки

$$l(i_k, j_k) = \min_{\substack{i,j \\ i,j \notin \{i_1, j_1, \ldots, i_{k-1}, j_{k-1}\}}} \{l(i, j)\}, \quad k = \overline{1, s}\ . \qquad (3.8)$$

Для кожних двох пар елементів цієї множини обов'язковим є виконання умов

$$i_k \neq i_l, \ \ i_k \neq j_l, \ \ i_k \neq i_l + 1, \ \ i_k \neq j_l + 1, \quad k, l \in \{1, 2, \ldots, s\}\ .$$

Процес продовжується доти, поки не буде отриманий порядок елементів, що відповідає одному з отриманих раніше.

Авторами [20] було досліджене зведення задачі комівояжера до задачі лінійного програмування.

Для складання математичної моделі задачі вводяться двійкові змінні

$$x_{ij} = \begin{cases} 1, & \text{якщо в маршруті присутній переїзд з} \\ & \text{пункту } i \text{ в пункт } j; \\ 0 & - \quad \text{в протилежному випадку.} \end{cases}$$

Крім того, уводяться цілочислові змінні u_i, u_j $\left(i, j = \overline{1, n}\right)$, що приймають довільні значення.

Математична модель задачі має вигляд:

$$\sum_{i=0}^{n} \sum_{j=0}^{n} d_{ij} x_{ij} \to \min_{x_{ij} \in \Omega} , \tag{3.9}$$

$$\Omega : \quad \sum_{\substack{i=0 \\ i \neq j}}^{n} x_{ij} = 1, \;\; j = \overline{1, n}; \tag{3.10}$$

$$\sum_{\substack{j=0 \\ j \neq i}}^{n} x_{ij} = 1, \;\; i = \overline{1, n} , \tag{3.11}$$

$$x_{ij} + x_{ji} \le 1, \;\; i, j = \overline{0, n}, \; i \neq j , \tag{3.12}$$

$$u_i - u_j + n x_{ij} \le n - 1, \;\; i, j = \overline{1, n}, \;\; i \neq j . \tag{3.13}$$

За прийнятний час за допомогою методів лінійного програмування цю задачу можна вирішити тільки при кількості пунктів транспортної мережі $n \le 10$. Більш того, при вирішенні задачі на ЕОМ, виникає необхідність округлення шуканих величин. Це ускладнює розв’язання та спричиняє спотворення результату [21].

До задачі цілочислового лінійного програмування може бути зведена також і задача розвезення [20]. Для одержання рішення задачі розвезення система рівнянь (3.10) – (3.13), що використовується для розв'язання задачі комівояжера, доповнюється обмеженнями, що забороняють побудову пересічних маршрутів. Це дозволяє знизити кількість необхідних обчислень і, отже, збільшити вимірність задачі, яка може бути вирішена за прийнятний час.

Через те, що застосування точних методів можливо тільки для задач невеликої вимірності (30 – 40 пунктів), більш широке застосування для розв'язання задачі розвезення здобули наближені методи.

Наближені методи пошуку рішення для задачі розвезення діляться на три групи: випадкового пошуку, локальної оптимізації та евристичні [5].

До методів випадкового пошуку відносять мікрорайонування клієнтів [22] та виділення типових ситуацій [23]. Розробка даної групи методів була обумовлена наявністю нестабільності попиту в пунктах завезення. У зв'язку з нестабільністю попиту виникає нерівномірність об'ємів і періодичності завезень, тоді як для постачальних організацій і автотранспорту бажано мати стабільні плани роботи.

Зазначені методи припускають поділ клієнтів (пунктів завезень) на групи в залежності від регулярності замовлень, що надходять від них. Для виділених типових ситуацій формується декілька можливих варіантів маршруту. Після одержання замовлення диспетчером здійснюється вибір найбільш відповідного з планів або коректування попередньо сформованих маршрутів.

Найбільш відомим методом локальної оптимізації є метод інверсій [24]. Основна ідея цього методу полягає в тому, що наявне початкове рішення розбивається на r фрагментів, що потім знову об'єднуються в один маршрут, але за допомогою ланок, відсутніх у початкову маршруті. При переборі всіх варіантів заміни r вилучених ребер, що забезпечують зменшення довжини маршруту, можна одержати такий, що не може бути поліпшений за допомогою цієї процедури. Такий маршрут називається r-оптимальним.

У [25] наведене дослідження 2-оптимальної процедури. За умови $r = 2$ для відновлення маршруту існують тільки дві ланки й один варіант об'єднання маршруту. При цьому один із фрагментів маршруту буде пройдений у протилежному напрямку. Якщо з маршруту вилучаються ланки $(i, i+1)$ і $(j, j+1)$ при $(i+1) < j$, то відновити маршрут можна шляхом включення в нього ланок (i, j) і $(i+1, j+1)$. При цьому скорочення довжини маршруту від такої операції оцінюється виразом

$$f_{i-j} = \sum_{k=i+1}^{j-1} d_{k,k+1} + d_{i,i+1} + d_{j,j+1} - \sum_{k=i+1}^{j-1} d_{k+1,k} - d_{i,j} - d_{i+1,j+1} \,. \tag{3.14}$$

Вибір значення r, рівного 2, гарантує найменші витрати часу для розв’язання задачі, однак отримане рішення перевищує оптимальне на 4,5% [24].

Відповідно до [24], при виборі $r = 3$ або $r = 4$ отримане рішенні відрізняється від оптимального на 1,1%. Однак при цьому значно збільшується об'єм обчислень. Якщо для 2-оптимальних планів на кожному кроці необхідно аналізувати $n(n-3)$ варіантів заміни ланок, то для 3-оптимальних планів – уже $4n(n-4)(n-5)/3$ варіантів. За оцінкою [5] метод інверсій для 3-оптимальних планів вимагає в 4 рази більш трудомістких обчислень, ніж, наприклад, метод Кларка-Райта.

Для скорочення кількості розглянутих варіантів у роботі [26] було запропоновано накладати додаткові обмеження при заміні.

Нехай при реалізації методу здійснюється заміна ребер (i_k, j_k) на ребра (l_k, m_k), $k = \overline{1, r}$. Тоді перестановка виконується тільки за умови виконання умови

$$\sum_{k=1}^{r} \left(d_{i_k, j_k} - d_{l_k, m_k} \right) > 0 \ . \tag{3.15}$$

У [27] зазначений спосіб, що поліпшує отриманий 3-оптимальний план за допомогою перестановки пунктів усередині маршрутів і між маршрутами. При виконанні цих перестановок вигода оцінюється виразом

$$f_{i-j} = d_{i-1,j} + d_{i,i+1} + d_{j,j+1} - d_{i-1,i+1} - d_{i,j+1} - d_{j,i} \ , \tag{3.16}$$

де i – порядковий номер на маршруті пункту, що переставляється; j – порядковий номер на маршруті пункту, після якого вставляється i-й пункт.

Найбільш поширеними методами розв’язання задачі розвезення є евристичні методи, що можна розділити на три групи. До першої групи відносяться методи, що моделюють дії досвідченого диспетчера – планувальника маршрутів; до другого – реалізуючі евристику (формалізоване суб'єктивне поняття про «кращий» маршрут); до третього – точні методи розв’язання, що спрощують обчислення та/або скорочують тривалість розрахунків ціною відмови від гарантій знайти точне рішення.

Одним із методів першої групи є мітелковий метод [28]. Метод використовує графічне подання транспортної мережі. Визначається розташування кожного i-го пункту, $i = \overline{2, n}$, відносно першого пункту у полярних координатах, тобто знаходиться довжина радіуса-вектора r_i і кут

$$An_i = arctg\frac{y_i - y_1}{x_i - x_1}\ , \tag{3.17}$$

де x_1, y_1 и x_i, y_i – декартові координати відповідно 1-го та i -го пунктів.

Початковий план формується на підставі ранжування пунктів за розміром кута An_i з урахуванням вантажопідйомності доступних транспортних засобів. Далі здійснюються перестановки пунктів між маршрутами з метою мінімізації сумарної довжини маршруту.

Мітелковий метод досить простий за алгоритмом обчислення, але якість отриманих рішень низька. Тому він зазвичай використовується в сполученні з методами локальної оптимізації, дозволяючи дуже швидко формувати шукане рішення [22, 24, 26]. Існує також метод найбільшого кута, який за способом подання транспортної мережі та іншими властивостями є близьким до розглянутого [29].

Частіше усього для розв’язання задачі розвезення застосовується метод Кларка-Райта [30]. На підставі даного методу початкова система маршрутів перетворюється таким чином, щоб кожне окреме перетворення давало найбільше поліпшення. У якості показника поліпшення Кларком і Райтом прийнята економія пробігу. Для її оцінки обчислюється матриця економії, елементи якої обчислюються за виразом

$$s_{ij} = d_{0,i} + d_{0,j} - d_{i,j}\ , \tag{3.18}$$

де $d_{0,i}$ – відстань між постачальником і пунктом i, $d_{0,j}$ – відстань між постачальником і пунктом j, $d_{i,j}$ – відстань між пунктами i і j.

На кожному кроці робиться об'єднання двох маршрутів. Після цього з матриці викреслюються ті економії, що вже не можуть бути реалізовані. Процес припиняється, коли всі позитивні економії викреслені.

У [10] наведено ще декілька способів розрахунку економії, що запропоновані Т.Гаскелем:

$$\pi_{ij} = S_{ij} - d_{i,j} \; ; \tag{3.19}$$

$$\lambda_{ij} = S_{ij}\left(d + \left| d_{0,i} - d_{0,j} \right| - d_{i,j}\right) , \tag{3.20}$$

де d – середня відстань від постачальника до пунктів завезення;

$$M_{ij} = S_{ij} - \Theta d_{i,j} \; , \tag{3.21}$$

де Θ – параметр структуризації маршруту, оптимальне значення якого визначається для кожного конкретного району перевезень шляхом зіставлення рішень, отриманих при різних значеннях Θ.

Т. Гаскелем були також розглянуті послідовний і рівнобіжний способи складання маршрутів. Послідовний спосіб припускає, що побудова нового маршруту починається тільки тоді, коли вичерпані можливості поліпшення попереднього або коли цей маршрут уже не можна нарощувати. При рівнобіжному способі всі маршрути нарощуються одночасно.

Порівняння різних способів розрахунку економії та способів побудови маршрутів показало, що кращий результат дає рівнобіжний спосіб при розрахунку економії за формулою (3.18) [31], хоча і це рішення може бути поліпшено методами локальної оптимізації [5]. У зв'язку з простотою обчислень цей метод працює досить економно, що й обумовило широке поширення його різноманітних версій [32, 33].

Найбільш економічні системи маршрутів можуть бути отримані за допомогою точних методів. Однак усі вони пред'являють високі вимоги до пам'яті ЕОМ, тому не можуть бути застосовані для розв'язання задач маршрутизації у транспортних мережах великої вимірності. На теперішній час точні методи спроможні забезпечувати ефективне рішення тільки для мереж, що містять не більш ніж 40 – 50 пунктів доставки.

У зв'язку з проблемою великої вимірності більш широке поширення одержали наближені методи. Методи, що засновані на суб'єктивній оцінці «кращого» маршруту або на спробі змоделювати дії досвідченого диспетчера, мають досить прості алгоритми обчислення та не вимагають занадто великих витрат пам'яті ЕОМ. Однак часто такі методи дають рішення, що сильно відхиляється від оптимального.

Більш стійкими до зміни умов задачі є наближені методи, що засновані на точних. У таких модифікаціях зменшення витрат пам'яті ЕОМ здійснюється за рахунок відмови від пошуку абсолютного оптимуму та зменшення числа розглянутих варіантів маршрутів. Часто, незважаючи на відсікання багатьох варіантів, такі методи приводять до одержання оптимального рішення. Але гарантій одержання такого рішення немає.

Таким чином, на цей час досить актуальною залишається розробка методів маршрутизації, що сполучать у собі високу точність і можливість їх застосування для транспортних мереж великої розмірності. Найбільш перспективними, на погляд авторів, є дослідження, що спрямовані на адаптацію точних методів до розв'язання задач великої вимірності.

3.2. Метод динамічного програмування та аналіз його ефективності

Запропонований авторами алгоритм розв’язання задачі комівояжера базується на класичному методі динамічного програмування, що викладається в [15]. Він дозволяє одержувати більш точні результати і вирішувати задачі більшої вимірності в порівнянні з класичним методом динамічного програмування.

Опишемо суть методу динамічного програмування стосовно до розв’язання задачі комівояжера.

Змістовна постановка задачі комівояжера така: *побудувати кільцевий маршрут мінімальної довжини, що починається в початковому пункті* 0, *проходить через* n *інших пунктів рівно по одному разу та закінчується в початковому пункті* 0.

3.2.1. Формування матриці відстаней між пунктами транспортної мережі

При розв’язанні задачі комівояжера прийнято вважати, що розглядається транспортна мережа, в якій існує шлях із кожного пункту в будь-який іншій. На практиці, якщо деякі два пункти не мають безпосереднього транспортного зв’язку, то для цих пунктів будується з'єднуючий маршрут мінімальної довжини, що проходить через деякі транзитні пункти.

Для прикладу розглянемо транспортну мережу одного із центральних районів м. Харкова, що містить 8 пунктів із двостороннім рухом на кожній ділянці маршрутів, що з'єднують ці пункти (рис. 3.1).

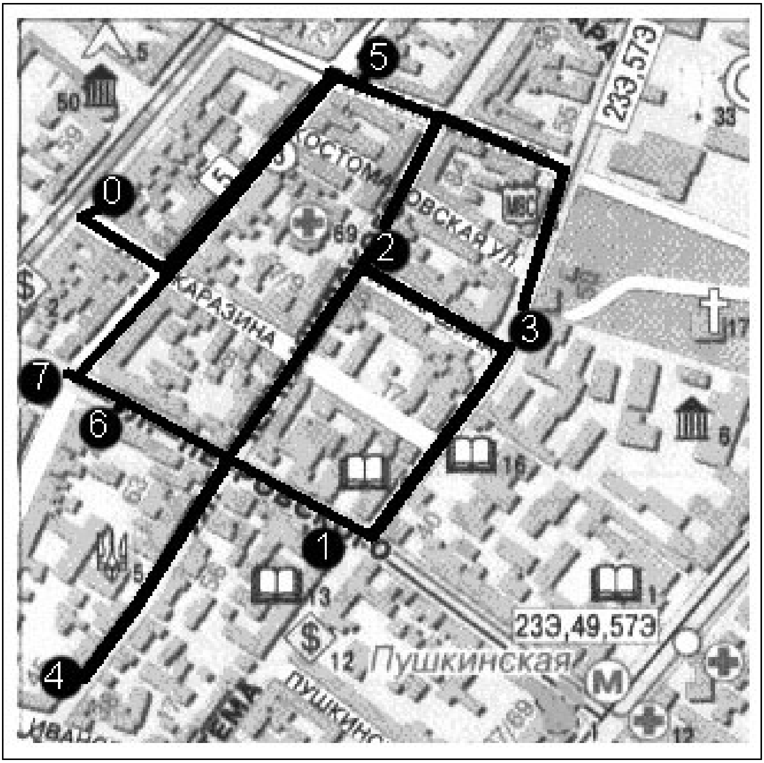


Рис. 3.1 – Схема транспортної мережі в умовах прикладу

У транспортній мережі на рис. 3.1 доцільно задати відстані між пунктами 0 і 5, 0 і 7, 1 і 2, 1 і 3, 1 і 4, 1 і 6, 2 і 3, 2 і 4, 2 і 5, 2 і 6, 3 і 5, 4 і 6, 5 і 7, 6 і 7, тому що між ними існує маршрут, що не проходить через будь-які інші пункти. Відповідно, між тими парами пунктів, що відсутні в наведеному списку, довжини шляхів задавати недоцільно. Вони визначаються шляхом нескладних обчислень.

В умовах прикладу відстані між пунктами з безпосереднім транспортним зв’язком однозначно подаються за допомогою матриці, що відповідає табл. 3.1.

Таблиця 3.1 – Довжини шляхів між пунктами транспортної мережі без транзитних пунктів

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*
0						406		288
1			452	245	495		390	
2		452		205	611	350	445	
3		245	205			500		
4		495	611				460	
5	406		350	500				447
6		390	445		460			48
7	288					447	48	

За умовами прикладу очевидно, що маршрут мінімальної довжини з пункту 0 у пункт 6 проходить через пункт 7. Його довжина дорівнює 288+48 = 336. Оптимальний маршрут із пункту 0 в 1 (0, 7, 6, 1) має довжину 288+48+390 = 726. Оптимальний маршрут із пункту 0 у пункт 4 також проходить через пункти 7 і 6 і має довжину 288+48+460 = 796. Уже не настільки очевидно, але теж досить просто можна визначити довжини оптимальних маршрутів із пункту 0 у 2 та з 0 у 3.

Для вибору маршруту з пункту 0 у пункт 2 розглянемо варіанти (0, 5, 2) та (0, 7, 6, 2). Довжини цих маршрутів відповідно дорівнюють 756 і 781. Довжина найкоротшого маршруту з 0 у 2 визначиться меншим із цих чисел, тобто 756. Аналогічним способом можна визначити, що довжина оптимального маршруту з пункту 0 у пункт 3 дорівнює 906, а самий маршрут відповідає послідовності пунктів (0, 5, 3).

Неважко одержати маршрути для інших пар пунктів, що не зв'язані між собою безпосередньо. У більш складних випадках використовуються спеціальні методи побудови найкоротших маршрутів між пунктами, наприклад, матричний.

У табл. 3.2 наведені усі відстані між пунктами транспортної мережі в умовах прикладу, тобто довжин шляхів між пунктами, що зв'язані безпосередньо, і довжини маршрутів між тими парами пунктів, для яких такий зв'язок відсутній.

Таблиця 3.2 – Відстані між пунктами транспортної мережі в умовах прикладу

	0	***1***	***2***	***3***	***4***	***5***	***6***	*7*
0		726	756	906	796	406	336	288
1	726		450	245	495	745	390	438
2	756	450		205	611	350	445	493
3	906	245	205		740	500	635	683
4	796	495	611	740		955	460	508
5	406	745	350	500	955		495	447
6	336	390	445	635	460	495		48
7	288	438	493	683	508	447	48	

Як видно на прикладі пунктів 1 і 2, більш вигідним із погляду довжини виявляється не маршрут, що безпосередньо зв'язує ці пункти, а маршрут, що проходить через пункт 3.

3.2.2. Алгоритм методу динамічного програмування стосовно розв’язання задачі комівояжера

Процес формування шуканого маршруту комівояжера здійснюється за n ітерацій, де n – кількість пунктів у транспортній мережі без урахування початкового (він же кінцевий) пункту в кільцевому маршруті комівояжера, тобто без нульового пункту.

На кожній ітерації формується n маршрутів, $M_1, M_2, ..., M_n$, кожний із який містить на 2 пункти більше, ніж номер ітерації. Маршрут $M_{k,i}$ на k-й ітерації, $i = \overline{1,n}$, $k = \overline{1,n-1}$, може бути описаний у такий спосіб. Він починається в пункті i, проходить через k відмінних від i пунктів і закінчується в пункті 0. Маршрут формується на підставі маршрутів, отриманих на попередній ітерації. У маршрути, що формуються, кожний із пунктів повинний входити не більш одного разу.

На першій ітерації для кожного значення i обчислюється функція, що оцінює довжину шляху з пункту i в кінцевий пункт 0 через пункт завезення j, $j \in \{1, ..., n,\ j \neq i\}$:

$$f(i;j) = \min_{j \in \{1,...,n,\ j \neq i\}} \{d_{ij} + d_{j0}\}, \tag{3.22}$$

де d_{ij} – відстань між пунктами i та j.

На рис. 3.2, 3.3 наведені схеми, що відображають відповідно першу та другу ітерації методу динамічного програмування для транспортної мережі, що зображена на рис. 3.1. Відстані між пунктами в цій транспортній мережі були наведені в табл. 3.2.

Для пункту 1 транспортної мережі на рис. 3.2 пунктирними лініями показані всі можливі маршрути, а жирною безперервною лінією показаний маршрут $M_{1,1} = (1,6,0)$, що обирається за критерієм (3.22):

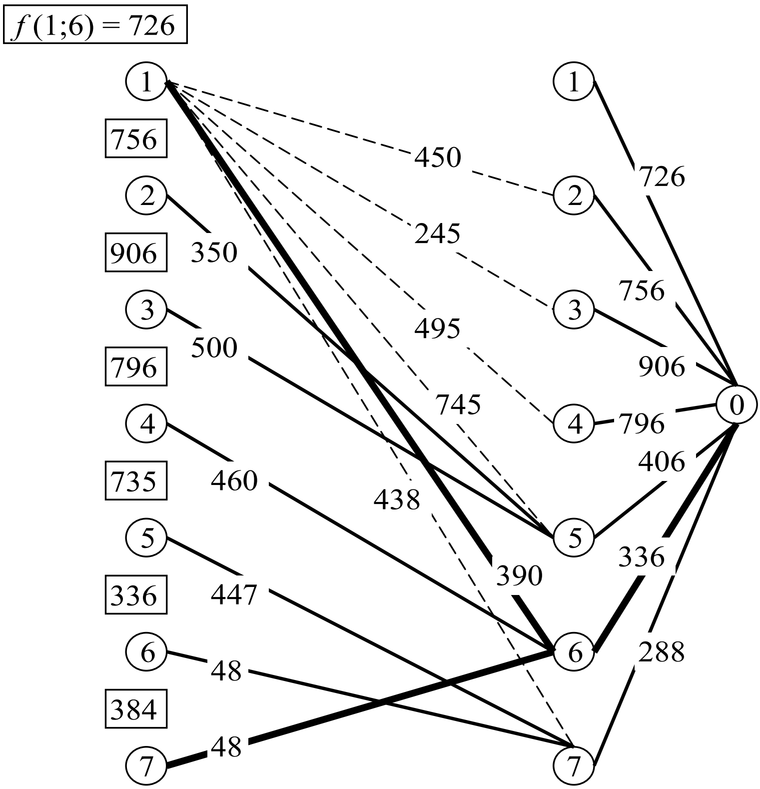


Рис. 3.2 – Схематичне зображення першої ітерації методу динамічного програмування для транспортної мережі в умовах прикладу

$$f(1;6)=$$

$$=\min\left\{\underbrace{450+756}_{(1,2,0)},\underbrace{245+906}_{(1,3,0)},\underbrace{495+796}_{(1,4,0)},\underbrace{745+406}_{(1,5,0)},\underbrace{390+336}_{(1,6,0)},\underbrace{438+288}_{(1,7,0)}\right\}=$$

$$=726\ .$$

Мінімальну довжину мають два останні розглянутих маршрути, $(1,6,0)$ і $(1,7,0)$, але відповідно до методу динамічного програмування повинний бути обраний тільки один.

На рис. 3.2 показані також мінімальні маршрути, що йдуть із пунктів 2, 3, …, 7 у пункт 0 і проходять через один транзитний пункт. У рамочці над кожним із пунктів указується значення функції (3.22), що визначають довжину маршруту:

$$f(2;5)=756\ ,\ \ f(3;5)=906,\ \ f(4;6)=796,$$
$$f(5;7)=735,\ f(6;7)=336,\ \ f(7;6)=384.$$

Критерій оптимальності маршруту з довільною кількістю транзитних пунктів також базується на функції, що визначає довжину шляху з пункту i в пункт 0:

$$f(i;\ j_1,\ j_2,\dots,j_k)=f(M_i)=$$
$$=\min_{1\le m\le k}\left\{d_{ij_m}+f\left(j_m;\ j_1,\ j_2,\dots,j_{m-1},j_{m+1},\dots,j_k\right)\right\}, \tag{3.23}$$

де $i=\overline{1,n}$, $\{j_1,j_2,\dots,j_k\}\subseteq\{1,2,\dots,i-1,i+1,\dots,n\}$.

Для прикладу знову розглянемо транспортну мережу, що зображена на рис. 3.1. На рис. 3.3 показана схема вибору в цій мережі маршруту з пункту 5 у пункт 0 із двома транзитними пунктами, тобто дії, що відносяться до другої ітерації розв'язання задачі.

Для пункту 5 довжина мінімального маршруту в пункт 0 визначається таким чином:

$$f(5;6,7)=\min\{745+f(1;6),\quad 955+f(4;6),$$
$$495+f(6;7),\quad 447+f(7;6)\}=831.$$

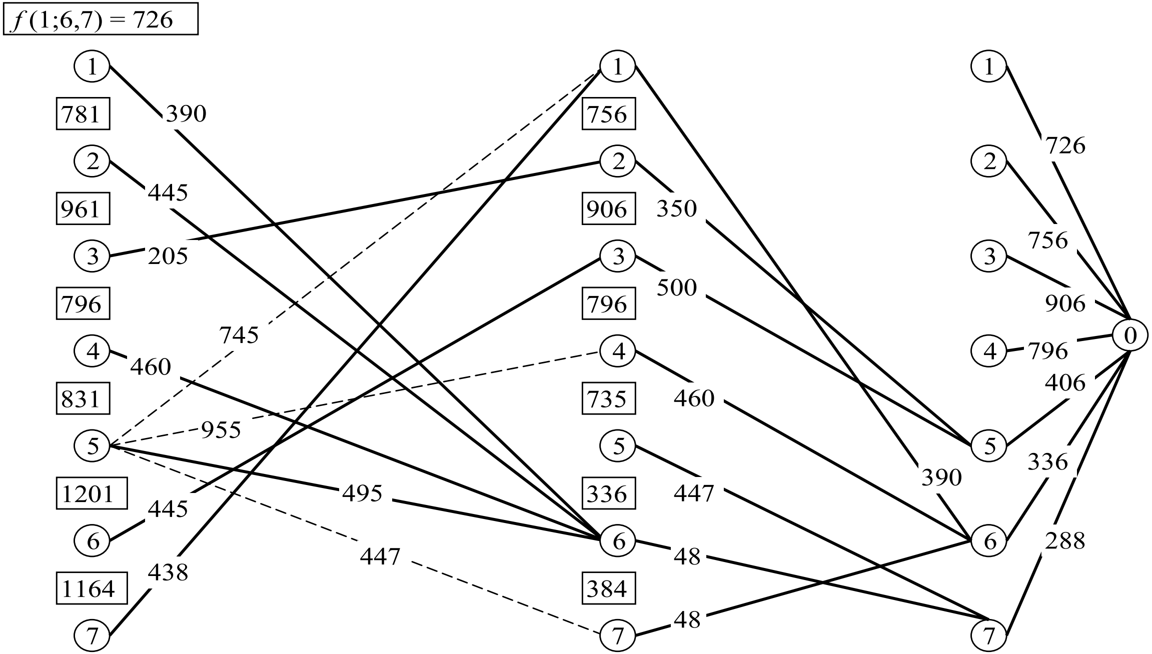


Рис. 3.3 – Схематичне зображення другої ітерації методу динамічного програмування для транспортної мережі в умовах прикладу

Таким чином, маршрут із пункту 5 у пункт 0 на другій ітерації визначається як $M_{2,5} = (5,6,7,0)$. Очевидно, що при виборі цього маршруту ми виключили з розгляду маршрути першої ітерації $M_{1,2}$ і $M_{1,3}$, тому що вони містять у собі пункт 5.

Може виникнути ситуація, коли на k-й ітерації буде неможливо побудувати маршрут, що починається в пункті i, на підставі результатів $(k-1)$-ї ітерації. Це відбувається тоді, коли всі маршрути $M_{k-1,1}, M_{k-1,2}, ..., M_{k-1,n}$ містять у собі пункт i: $M_{k-1,1}(i), M_{k-1,2}(i), \ldots, M_{k-1,n}(i)$.

Як приклад розглянемо знову транспортну мережу, що зображена на рис. 3.1.

При побудові маршрутів відповідно до методу динамічного програмування на 4-й ітерації маємо маршрути: $M_{4,1} = (1,7,6,2,5,0)$, $M_{4,2} = (2,3,1,6,7,0)$, $M_{4,3} = (3,2,1,6,7,0)$, $M_{4,4} = (4,1,3,2,5,0)$, $M_{4,5} = (5,3,1,6,7,0)$, $M_{4,6} = (6,1,3,2,5,0)$, $M_{4,7} = (7,1,3,2,5,0)$. Очевидно, що при переході до ітерації 5 на підставі даних маршрутів неможливо буде побудувати маршрут, що починається в пункті 1, тому що всі отримані на 4-й ітерації маршрути містять цей пункт.

У розглянутому випадку здійснюється повернення до ітерації $k-2$. На підставі маршрутів $M_{k-2,1}, ..., M_{k-2,n}$ будуються альтернативні маршрути $M'_{k-1,1}, ..., M'_{k-1,n}$, що не проходять через пункт i, і шуканий маршрут $M_{k,i}$ формується на підставі альтернативних маршрутів.

Для транспортної мережі в умовах приклада (рис. 3.1) на 3-й ітерації методу динамічного програмування маємо декілька маршрутів, що не проходять через пункт 1:

$M_{3,5} = (5,2,6,7,0)$, $M_{3,6} = (6,3,2,5,0)$, $M_{3,7} = (7,6,2,5,0)$, $f(M_{3,5}) = 1131$, $f(M_{3,6}) = 1596$, $f(M_{3,7}) = 1249$. Використовуючи ці маршрути, можна побудувати альтернативні маршрути: $M'_{4,3} = (3,5,2,6,7,0)$, $M'_{4,4} = (4,7,6,2,5,0)$, $f(M'_{4,3}) = 1631$, $f(M'_{4,4}) = 1757$.

Таким чином, застосовуючи результати 3-ї ітерації, може бути побудований маршрут $M_{5,1} = (1,3,5,2,6,7,0)$.

Якщо всі маршрути $(k-2)$-ї ітерації теж містять пункт i, то відбувається повернення до ітерації $k-3$, і так далі. Для цього протягом усієї реалізації методу необхідно зберігати результати, що отримані на всіх кроках від першого до останнього.

На останній ітерації з номером $k = n$ будується маршрут $M_{n,0}$, що починається в пункті 0, проходить через всі пункти $1,2,...,n$ і закінчується в пункті 0. Для цього серед маршрутів $M_{n-1,1},...,M_{n-1,n}$ вибирається той, для якого реалізується умова

$$f(M_{n,0}) = \min_{1 \le m \le n} \{d_{0m} + f(M_{n-1,m})\}, \tag{3.24}$$

або інакше

$$f(0;1,2,...,n) = \min_{1 \le m \le n} \{d_{0m} + f(m;1,2,...,m-1,m+1,...,n)\}. \tag{3.25}$$

У [15] стверджується, що метод динамічного програмування є оптимальним. Однак при порівнянні результатів, що отримані за допомогою цього методу, із результатами реалізації методу «гілок і границь» з'ясувалося, що «оптимальні» маршрути динамічного програмування в середньому на 2,9 %

перевищують маршрути, що отримані за допомогою методу «гілок і границь».

3.2.3. Тестування методу динамічного програмування стосовно розв'язання задачі комівояжера

Для тестування методів розв'язання задачі комівояжера в програмному середовищі *Delphi* 5 створені спеціальні програми, що реалізують метод «гілок і границь» і метод динамічного програмування.

У процесі тестування знайдені довжини маршрутів комівояжера для 40 різних графів із кількістю вершин від 8 до 18. Вершини вибиралися довільним способом, відстані між ними вимірювалися по прямій. Довжини маршрутів, що отримані за допомогою методу динамічного програмування, у ряді випадків виявлялися більше довжин маршрутів, отриманих за допомогою методу «гілок і границь». У табл. 3.3 наводиться середнє і максимальне відхилення довжин цих маршрутів при різній вимірності транспортних мереж.

Таблиця 3.3 – Відхилення довжин маршрутів, отриманих методом динамічного програмування, від отриманих методом «гілок і границь»

Тип відносного показника	Кількість пунктів в мережі		
	8 — 10	12 — 14	15 — 18
Середнє відхилення, %	2,3	1,5	4,2
Максимальне відхил., %	20,9	4,6	10

Більш докладні дані про транспортні мережі, для яких проводилося тестування, і результати самого тестування наводяться в табл. 3.4.

Таблиця 3.4 – Результати тестування методу динамічного програмування

№ задачі	Відстань між пунктами			Кількість пунктів	Довжина маршруту комівояжера		Відносне відхилення
	найменша	найбільша	середня		оптимальна	за методом ДП	
1	131	1073	522,32	8	2563	2580	0,0066
2	133	837	436,18	8	2441	2950	0,2085
3	48	955	544,68	8	2497	2497	0
4	157	1273	681,25	8	2980	2980	0
5	229	1405	800,11	8	3888	3888	0
6	167	761	422,66	8	2213	2213	0
7	66	874	400,89	8	2195	2203	0,0036
8	69	1698	724,64	8	3873	3873	0
9	43	1081	518,43	8	2676	2676	0
10	78	677	357,25	8	1797	1797	0
11	154	1225	700,14	8	2899	2899	0
12	159	913	480,89	8	2372	2690	0,1341
13	212	1012	592,68	8	3389	3596	0,0611
14	11	88	48,68	8	244	246	0,0082
15	0	75	39,25	9	192	192	0
16	1	65	28,97	9	157	157	0
17	2	45	20,36	9	141	141	0
18	8	98	49,84	10	264	270	0,0227
19	15	165	68,07	10	454	479	0,0551
20	11	108	58,4	10	323	335	0,0372

Закінчення табл. 3.4

№ задачі	Відстань між пунктами			Кількість пунктів	Довжина маршруту комівояжера		Відносне відхилення
	найменша	найбільша	середня		оптимальна	за методом ДП	
21	1	9	4,37	10	22	23	0,0455
22	1	28	13,87	10	97	98	0,0103
23	1	46	23,5	10	144	144	0
24	115	904	481,93	10	2619	2619	0
25	110	955	451,39	10	2490	2505	0,006
26	75	1080	490,22	10	2299	2299	0
27	5	57	29,12	12	231	238	0,0303
28	145	2128	866,5	12	4945	4945	0
29	10	76	40,2	12	212	212	0
30	21	262	112,56	12	653	683	0,0459
31	2	26	13,11	14	88	92	0,0455
32	1	55	18,88	14	163	163	0
33	0	45	17,08	14	162	162	0
34	0	49	25,23	14	209	209	0
35	1	42	16,89	15	166	171	0,0301
36	0	41	20,52	16	207	216	0,0435
37	13	153	74,53	16	513	513	0
38	2	99	44,42	16	513	513	0
39	4	117	61,22	18	471	507	0,0764
40	0	46	13,81	18	130	143	0,1

Крім того, що метод динамічного програмування не гарантує одержання точних результатів, він має ще два недоліки, а саме:

– по-перше, через необхідність зберігати всю проміжну інформацію від першої до останньої ітерації треба мати великий об'єм динамічної пам'яті ЕОМ;

– по-друге, у випадку повернення на попередні ітерації суттєво збільшується час вирішення задачі.

Останні два недоліки обмежують припустиму вимірність задачі, що можна розв'язати на ЕОМ за прийнятний час. Зокрема, при тестуванні програми, що реалізує алгоритм динамічного програмування на ЕОМ із процесором частотою 1,5 ГГЦ та оперативною пам'яттю 256 МБ, удавалося вирішувати задачу комівояжера тільки для транспортних мереж, що містять не більше 45 пунктів.

3.3. Модифікований метод динамічного програмування та оцінка його ефективності

Авторами пропонується модифікований метод динамічного програмування, що дозволяє, з одного боку, зменшити об'єм зберігання даних і кількість обчислень, а з іншого – підвищити точність результату.

При розробці модифікованого методу динамічного програмування було розглянуто декілька варіантів модифікації.

Для опису суті першого варіанта розглянемо ситуацію, коли на k-й ітерації не вдається побудувати маршрут $M_{k,i}$ на підставі маршрутів попередньої ітерації.

Приклад такої ситуації показаний у попередньому підрозділі для транспортної мережі, зображеної на рис. 3.1. На ітерації $k=5$ неможливо побудувати маршрут для пункту 1 на підставі маршрутів попередньої ітерації, тому що всі маршрути ітерації $(k-1)=4$ містять цей пункт.

У класичному методі динамічного програмування допускається побудова маршруту на підставі ітерацій $k-2,\ k-3,\ ...,\ 1$, якщо в цьому виникає необхідність.

Для зменшення витрат ресурсів ЕОМ можна ввести обмеження на кількість номерів ітерацій, до яких здійснюється повернення. При такому обмеженні з більшою ймовірністю з'являються пункти, для яких на окремих ітераціях стане неможливим побудувати маршрут.

Як приклад знову розглянемо транспортну мережу, що зображена на рис. 3.1. Як уже було сказано, при розв'язанні задачі комівояжера методом динамічного програмування на ітерації з номером $k=4$ були отримані маршрути $M_{4,1}=(1,7,6,2,5,0)$, $M_{4,2}=(2,3,1,6,7,0)$, $M_{4,3}=(3,2,1,6,7,0)$, $M_{4,4}=(4,1,3,2,5,0)$, $M_{4,5}=(5,3,1,6,7,0)$, $M_{4,6}=(6,1,3,2,5,0)$, $M_{4,7}=(7,1,3,2,5,0)$.

Припустимо, що при побудові маршрутів було застосовано обмеження: в процесі вирішення можна повертатися не більш ніж на одну ітерацію. Тобто при побудові маршрутів на k-й ітерації можна використовувати тільки маршрути, що були отримані на ітераціях $k-1$ і $k-2$.

Як уже було показано, для транспортної мережі на рис. 3.1 на 5-й ітерації при вище зазначеному обмеженні можна побудувати маршрути для всіх пунктів, в тому числі для пункту із номером 1. Наприкінці виконання 5-ї ітерації були отримані маршрути: $M_{5,1}=(1,3,5,2,6,7,0)$, $M_{5,2}=(2,5,3,1,6,7,0)$, $M_{5,3}=(3,1,7,6,2,5,0)$, $M_{5,4}=(4,2,3,1,6,7,0)$, $M_{5,5}=(5,2,3,1,6,7,0)$, $M_{5,6}=(6,7,1,3,2,5,0)$, $M_{5,7}=(7,6,1,3,2,5,0)$.

Очевидно, що всі маршрути, що були побудовані на ітерації $k=5$, містять пункт 1.

При виконанні ітерації $k=6$ з урахуванням зробленого обмеження можливо використовувати тільки маршрути, що отримані на ітераціях $(k-2)=4$ і $(k-1)=5$.

Таким чином, виявляється неможливим побудувати маршрут, що починається в пункті 1, проходить через 6 пунктів і закінчується в пункті 0.

Оцінимо точну верхню границю $Q_м$ кількості маршрутів , що треба дослідити. На останній n-й ітерації до всіх побудованих маршрутів приєднується пункт 0. Точна верхня границя кількості маршрутів, що переглядаються на даній ітерації, дорівнює n ($Q_{м(n)}=n$). Кількість маршрутів, що переглядається на всіх інших ітераціях, залежить від обмежень на повторний аналіз маршрутів попередніх ітерацій. Таким чином, точна верхня границя загальної кількості маршрутів визначиться виразом

$$Q_м = Q_{м(1)} + Q_{м(2)} + \ldots + Q_{м(n-1)} + Q_{м(n)} = \sum_{k=1}^{n-1} Q_{м(k)} + n \ . \tag{3.5}$$

Оцінимо числа $Q_{м(1)}$, $Q_{м(2)}$, ..., $Q_{м(n-1)}$, що обмежують кількість досліджуваних маршрутів на ітераціях $1,\ 2,\ \ldots,\ n-1$ при різних обмеженнях на використання попередніх ітерацій.

При можливості використання на k-й ітерації тільки маршрутів $(k-1)$-ї ітерації при виборі маршруту для кожного з n пунктів досліджуються не більш ніж $n-1$ маршрут попередньої ітерації. Це число однакове для будь-якої ітерації, тобто

$$Q_{м(k)} = n-1\ , \quad k=1,2,\ldots,n-1\ . \tag{3.6}$$

Таким чином, за всю реалізацію алгоритму обчислюються і порівнюються довжини не більш ніж

$$Q_м = n(n-1)^2 + n \tag{3.7}$$

маршрутів із різною кількістю пунктів.

Припустимо, що на k-й ітерації можливе використання маршрутів $(k-2)$-ї ітерації. У цьому випадку побудова починається з аналізу маршрутів ітерації $k-1$. Для кожного з n пунктів аналізується $n-1$ маршрут. Після цього може виникнути необхідність побудови альтернативних маршрутів на підставі $(k-2)$-ї ітерації. Нехай така необхідність виникнула для пункту i. Альтернативні маршрути ми будемо будувати для пунктів 1, 2,...., $i-1$, $i+1$, ..., n. Загальна кількість цих пунктів дорівнює $n-1$. Для кожного такого пункту j (із числа зазначених) доцільно проаналізувати $n-2$ маршрути, тому що інші два із загальної кількості маршрутів обов'язково починаються з пунктів i або j. Маємо $(n-1)(n-2)$ маршрутів, що аналізуються на ітерації $(k-2)$. Після цього для пункту i необхідно буде знову розглянути щонайбільше $n-1$ маршрут. Отже, для пункту i може бути переглянуто $(n-1)(n-2)+(n-1)=(n-1)^2$ маршрутів.

Число пунктів, для яких потрібна побудова альтернативних маршрутів, може бути неоднаковим для різних ітерацій. Велика ймовірність того, що на початкових ітераціях не буде жодного такого пункту, а на заключних їх буде декілька. Для наведеної на рис. 3.1 транспортної мережі на ітераціях 1 – 4 таких пунктів не було. На 5-й ітерації альтернативні маршрути треба було будувати для одного пункту, на 6-й – для трьох. Число $(n-1)^2$ є явно завищеним, тому що далеко не для кожного з $(n-1)$ пунктів можна побудувати альтернативний маршрут.

Будемо вважати, що в середньому на кожній ітерації необхідно побудувати $(n-1)$ альтернативних маршрутів для одного пункту. Тоді верхня границя кількості маршрутів, що аналізуються на k-й ітерації, складе

$$Q_{м(k)} = (n-1)n + (n-1)^2(n-2) \; . \tag{3.8}$$

З урахуванням того, що загальне число ітерацій дорівнює $(n-1)$, орієнтовно здійснюється перегляд не більш

$$Q_м = (n-1)^3(n-2) + n(n-1)^2 + n \tag{3.9}$$

маршрутів.

При скороченні кількості ітерацій, до яких припускається повернення, знижується також і точність отриманих результатів. Тому для підвищення точності пропонується поряд із зменшенням кількості даних, що зберігаються, збільшити кількість аналізованих варіантів на невеликій кількості ітерацій.

Опишемо k-у ітерацію запропонованої модифікації для підвищення точності отриманого рішення.

Нехай необхідно вибрати маршрут мінімальної довжини, що починається в пункті i і закінчується в пункті 0. Як і в методі динамічного програмування, до моменту виконання цієї ітерації були побудовані і збережені маршрути $M_{1,1}$, ...,$M_{1,n}$; $M_{2,1}$, ..., $M_{2,n}$; ...; $M_{k-1,1}$, ..., $M_{k-1,n}$. Деякі з цих маршрутів містять пункт i, інші його не містять. Припустимо, маршрути M_{k-1,j_1}, ..., M_{k-1,j_l} містять у собі пункт

i, причому $j_1, ..., j_l \neq i$. Побудуємо альтернативні маршрути $M'_{k-1,j_1}, ..., M'_{k-1,j_l}$, для яких виконується умова

$$f\left(M'_{k-1,j_g}\right) = \min_{\substack{1 \le h \le n \\ h \neq j_g \\ i \notin M_{k-2,h}}} \left\{d_{j_h h} + f\left(M_{k-2,h}\right)\right\}. \qquad (3.10)$$

Далі, розглянемо маршрути, що отримані на ітерації $(k-2)$. Замінимо ті з них, у яких зустрічається пункт i, маршрутами мінімальної довжини, у яких пункт i відсутній. При цьому маршрут із мінімальною довжиною визначаємо за допомогою виразу типу (3.10). Використовуючи отримані маршрути, побудуємо ще один набір альтернативних маршрутів $M''_{k-1,1}, ..., M''_{k-1,i-1}, M''_{k-1,i+1}, ..., M''_{k-1,n}$. Повторюємо ту ж операцію для ітерацій $k-3$, $k-4$ і так далі. Після цього вибираємо маршрут найменшої довжини, отриманий при приєднанні пункту i до одного з побудованих маршрутів.

При поверненні до всіх попередніх ітерацій, у тому числі до першої, кількість обчислень за цим модифікованим методом динамічного програмування має той же порядок, що і кількість обчислень за методом повного перебору. Для скорочення обчислень пропонується обмежитися поверненням *не більш ніж на 4 ітерації*. При цьому кількість обчислень не перевищить значення поліному від n порядку п'ятого ступеня.

Розроблено декілька модифікацій методу динамічного програмування, що у порівнянні з класичним методом динамічного програмування забезпечують розв'язання задачі комівояжера для транспортних мереж значно більшої вимірності.

У запропонованих модифікаціях закладений параметр, що дозволяє при розв'язанні конкретних задач робити вибір на користь більшої вимірності транспортних мереж або більшої точності рішення. У будь-якому випадку використання модифікацій приводить до більш якісних рішень, ніж класичний метод.

3.4. Тестування модифікованого методу динамічного програмування

Для виявлення точності отриманих рішень і максимально припустимої вимірності в задачах комівояжера, що розв'язуються запропонованою модифікацією методу динамічного програмування, було проведено відповідне тестування. Тестування робилося на ЕОМ із процесором частотою 1,5 ГГЦ і оперативною пам'яттю 256 МБ. При тестуванні були складені маршрути комівояжера для 40 різних задач вимірністю від 8 до 18 пунктів (табл. 3.4).

У табл. 3.5 наведені результати тестування що дозволяють порівняти модифікації методу динамічного програмування з методом «гілок і границь» за точністю отриманих рішень.

У табл. 3.6 наведені максимальні вимірності задач, що удавалося вирішити на ЕОМ із зазначеними вище характеристиками за прийнятний час. Відстані між пунктами вибиралися випадковим способом.

В усіх таблицях використовуються наступні позначення для модифікацій методу динамічного програмування:

МДП1 – модифікований метод динамічного програмування з поверненням не більш ніж на 1 крок для *недосяжних за результатами попередньої ітерації* пунктів;

МДП2 – модифікований метод динамічного програмування з поверненням не більш ніж на 2 кроки для *недосяжних за результатами попередньої ітерації* пунктів;

Таблиця 3.5 – Відхилення результатів, отриманих за допомогою модифікацій методу динамічного програмування, від результатів, отриманих за допомогою методу «гілок і границь»

Модификація методу	МДП1	МДП2	МДП3	МДП4	МДП5	МДП6	МДП7	МДП8	МДП9	МДП10
Середнє відхилення, %	5,02	3,97	3,59	3,11	4,39	3,09	2,73	1,2	1,62	1,81
Максимальне відхилення, %	22,9	22,85	22,85	21,84	23,73	16,85	20,85	10	10	14,86

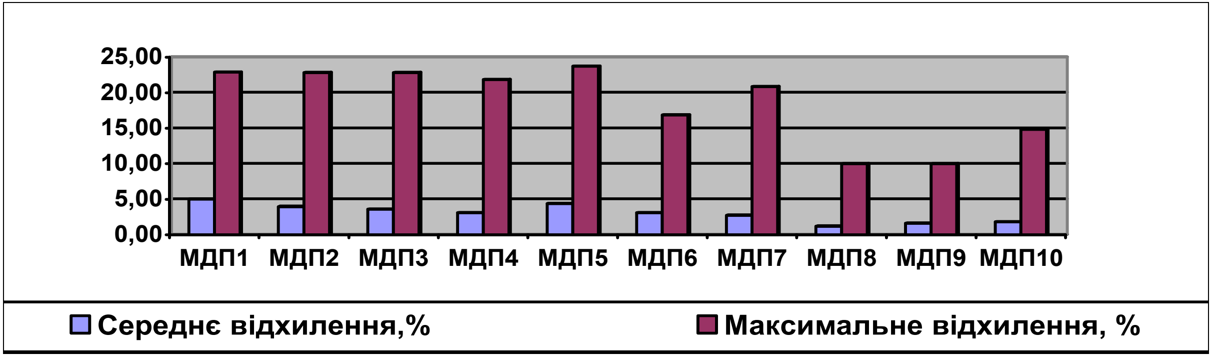


Рис. 3.3 Діаграма відносних показників відхилення результатів розв’язання задачі комівояжера за допомогою модифікацій методу динамічного програмування від оптимального рішення

Таблиця 3.5 – Максимальна кількість пунктів у задачах, що удалося вирішити за прийнятний час при тестуванні

Модификація методу	МДП1	МДП2	МДП3	МДП4	МДП5	МДП6	МДП7	МДП8	МДП9	МДП10
Максимальна кількість пунктів в задачі	120	110	95	80	100	95	90	80	75	65

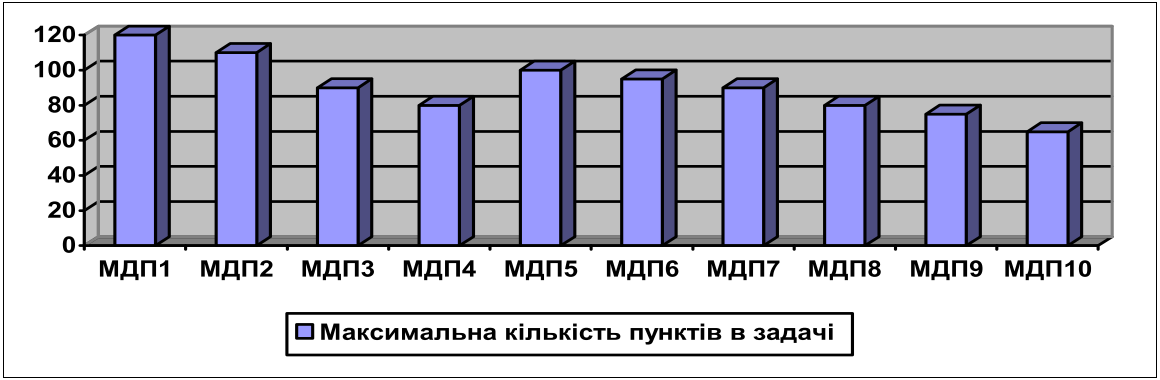


Рис. 3.4 – Діаграма припустимої вимірності задачі комівояжера для різних модифікацій методу динамічного програмування

МДП3 – модифікований метод динамічного програмування з поверненням не більш ніж на 3 кроки для *недосяжних за результатами попередньої ітерації* пунктів;

МДП4 – модифікований метод динамічного програмування з поверненням не більш ніж на 4 кроки для *недосяжних за результатами попередньої ітерації* пунктів;

МДП5 – модифікований метод динамічного програмування з поверненням на 1 крок для *всіх* пунктів;

МДП6 – модифікований метод динамічного програмування з поверненням на 2 кроки для *всіх* пунктів;

МДП7 – модифікований метод динамічного програмування з поверненням на 3 кроки для *всіх* пунктів;

МДП8 – модифікований метод динамічного програмування з поверненням на 4 кроки для *всіх* пунктів;

МДП9 – модифікований метод динамічного програмування з поверненням на 5 кроків для *всіх* пунктів;

МДП10 – модифікований метод динамічного програмування з поверненням на 6 кроків для *всіх* пунктів.

Порівняльний аналіз показує, що найбільш вигідним з погляду точності є модифікований метод динамічного програмування з поверненням на 4 кроки для всіх пунктів (МДП8 у таблицях). Ця модифікація дозволяє вирішувати задачі досить великої вимірності. Модифікований метод динамічного програмування з поверненням на 3 кроки для всіх пунктів, забезпечує ту ж точність, що і метод динамічного програмування, але в той же час дозволяє вирішувати задачі з подвоєним числом пунктів. При тестуванні методу «гілок і границь», що на теперішній час вважається самим точним і поширеним, з'ясувалося, що на ЕОМ із вище зазначеними технічними параметрами можна вирішити задачу комівояжера для транспортної мережі, у якій не більше 25 пунктів.

Модифікований метод динамічного програмування з поверненням не більш ніж на 1 крок для недосяжних за результатами попередньої ітерації пунктів (модифікація МДП1) дозволяє вирішувати задачі для транспортних мереж найбільшої вимірності. Цим методом можна скористатися при необхідності отримати грубу оцінку довжини маршруту комівояжера для транспортної мережі великої вимірності.

Треба зазначити, що модифікований метод динамічного програмування з поверненням не більш ніж на 1 крок для недосяжних за результатами попередньої ітерації пунктів (модифікація МДП1) корисний для використання в якості складового в алгоритмах, у яких присутнє багатократне розв’язання задачі комівояжера як частина розв’язання загальної задачі.

Те ж саме можна сказати про модифікований метод динамічного програмування з поверненням не більш ніж на 1 крок для всіх пунктів (модифікація МДП5).

Таким чином, запропоновану модифікацію методу динамічного програмування МДП5 можна вважати досить точною і придатною для розв’язання задачі комівояжера великої вимірності. Ця модифікація дозволяє вирішувати задачі великої вимірності за рахунок відмови від високої точності, і навпаки, задавати більш високу точність рішення для транспортних мереж із порівняно невеликою кількістю пунктів.

* * *

Алгоритми запропонованих модифікацій методу динамічного програмування реалізовані у вигляді єдиної програми Traveling_Salesman_DP.exe. У якості вихідних даних виступають відстані між пунктами транспортної мережі. Як і у випадку із задачею розбиття транспортної мережі на райони, необов'язково, щоб були задані довжини маршрутів між кожною парою пунктів. Для реальних транспортних мереж, у яких відстані задані не для всіх пар пунктів, у програмі передбачена опція «Оптимізувати», за допомогою якої при необхідності обчислюються довжини найкоротших маршрутів між пунктами. Побудова маршрутів реалізовано з використанням матричного методу.

У якості результату програма видає найменшу довжину маршруту, що проходить через усі пункти мережі рівно по одному разу, та послідовність пунктів у маршруті. Програма також дозволяє простежити послідовність побудови маршрутів, використовуючи покрокове виконання алгоритму. Більш докладно програма *Traveling_Salesman_DP.exe* буде описана в розділі 4.

РОЗДІЛ 4

ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ ДЛЯ ВИЗНАЧЕННЯ МАРШРУТУ КОМІВОЯЖЕРА ТА РОЗБИТТЯ ТРАНСПОРТНОЇ МЕРЕЖІ НА РАЙОНИ

В цьому розділі описуються програми *SlsManDProg.exe* і *Partition.exe*, що реалізують модифікації відповідно методу динамічного програмування для рішення задачі комівояжера великої вимірності і методу «гілок і границь» для розбиття транспортної мережі на райони. Названі програми є практичною реалізацією алгоритмів описаних у другому і третьому розділах монографії.

Результати тестування програми *SlsManDProg.exe* дозволяють судити про середню точність рішення, що можна одержати за допомогою розробленої модифікації методу динамічного програмування. На підставі проведеного тестування можна зробити висновок, що розроблена модифікація методу динамічного програмування може бути застосована для розв'язання задач комівояжера у транспортних мережах великої вимірності. Висока швидкодія програми *SlsManDProg.exe* дозволяє застосовувати її як підпрограму в реалізаціях алгоритмів, для яких розв'язання задачі комівояжера є проміжною дією.

Результати тестування програми *Partition.exe* дають можливість зробити висновок про властивості розбивки транспортної мережі на райони, отриманої за допомогою розробленої модифікації методу «гілок і границь».

Після виконання розбиття транспортної мережі на райони за допомогою програми *Partition.exe* відразу стає відома довжина маршруту комівояжера, що дозволяє судити про грошові та тимчасові витрати на транспортні перевезення в цьому районі.

Використання для оцінки вимірності районів довжини маршруту комівояжера дозволяє судити про площу, що займає район, а співвідношення довжин маршрутів комівояжера – про співвідношення площ цих районів.

4.1. Програмна реалізація модифікованого методу динамічного програмування для розв'язання задачі комівояжера

Програма призначена для побудови маршруту комівояжера мінімальної довжини в довільно поданій транспортній мережі. Цю програму доцільно використовувати при визначенні маршруту комівояжера для транспортної мережі з кількістю пунктів понад 25 (для більш потужних сучасних комп'ютерів – понад 50 пунктів). Верхня границя застосовності програми також залежить від потужності комп'ютера та може становити 170 і навіть 200 пунктів.

Як відомо, маршрут комівояжера проходить через усі пункти транспортної мережі рівно один раз та є кільцевим, тобто враховує також повернення з останнього пункту в перший, із якого маршрут починався.

Назва програми – *SlsManDProg.Exe*.

Програма написана в програмному середовищі *Delphi* 5 і призначена для роботи під операційною системою Windows-95 (-98, -XP, -*NT* -2000 і т.д.). Ніякого додаткового програмного забезпечення для запуску та виконання програми не потрібно. Редагування та рекомпіляція програми виконується в середовищі програмування *Delphi* 5 і більш пізніх її версіях.

За допомогою програми *SlsManDProg.exe* будується маршрут комівояжера мінімальної довжини та визначається його довжина. Програма припускає варіювання параметрами. Це дозволяє збільшити припустиму вимірність транспортної мережі за рахунок відмови від точності. Найменше середнє відхилення шуканого маршруту, отриманого в результаті роботи програми, становить 1,2% від оптимального маршруту. З такою точністю отримують рішення для транспортної мережі, що містить не більше 80 пунктів.

Програма дозволяє будувати маршрут комівояжера для транспортної мережі, що містить понад 170 пунктів. Однак у цьому випадку відхилення отриманого рішення від оптимального в середньому становить 7%.

Побудова оптимального маршруту здійснюється програмою відповідно до алгоритму модифікації методу динамічного програмування, що був описаний в третьому розділі цієї монографії.

Задача комівояжера розв’язується тільки за умови, якщо в транспортній мережі є можливість побудувати маршрут між будь-якими двома пунктами. Граф, побудований на підставі такої транспортної мережі, повинний бути зв’язним.

У якості вихідних даних беруться довжини шляхів між парами пунктів транспортної мережі, для яких існує безпосередній транспортний зв'язок. У випадку, якщо транспортна мережа така, що безпосередній транспортний зв'язок існує не для усіх пар пунктів, програма забезпечує побудову маршруту мінімальної довжини для кожної такої пари пунктів.

Відстані між пунктами транспортної мережі повинні бути подані цілими числами. Середня відстань між парами пунктів не повинна перевищувати числа $2147473647/(n + 1)$, де n – кількість пунктів транспортної мережі.

Алгоритм програми:

1) Запуск програми

Після запуску програми на екрані дисплея з'являється основне вікно програми. Вигляд вікна показано на рис. 4.1. На ньому є присутнім головне меню, а також закладки *Матриця відстаней*, *Маршрути*, *Маршрут повернення*, *Збережені кроки, Оптимальний маршрут*. Закладки являють собою сторінки (*Pages*) об'єкту *PageControl*. На кожній із закладок знаходяться одна або дві таблиці.

У таблиці на закладці *Матриця відстаней* зберігаються дані про довжину шляху між пунктами транспортної мережі.

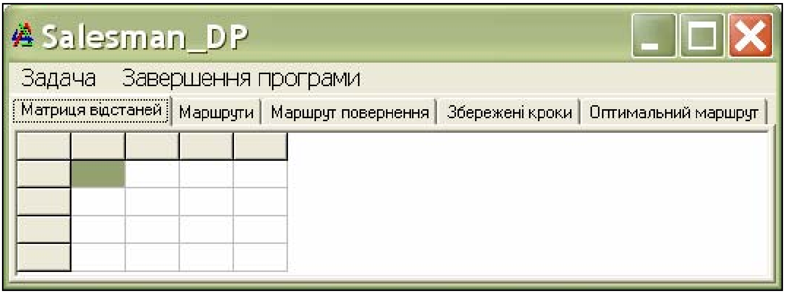


Рис. 4.1 – Основне вікно програми *SlsManDProg.exe*

У таблиці на закладках *Маршрути* і *Маршрут повернення* заносяться проміжні маршрути, що отримують в результаті реалізації модифікованого методу динамічного програмування.

У таблиці на закладці *Збережені кроки* запам'ятовуються маршрути, що генеруються на кожній ітерації.

На закладці *Оптимальний маршрут* містяться кінцеві результати – довжина маршруту комівояжера найменшої довжини та послідовність пунктів у маршруті.

2) Виконання процедур, що запускаються з головного меню основного вікна програми

Виконання процедур для розв'язання задачі комівояжера рекомендується здійснювати в наступній послідовності:

2.1) Спочатку для завантаження параметрів нової транспортної мережі треба вибрати в головному меню пункт *Задача/Нова* (рис. 4.2).

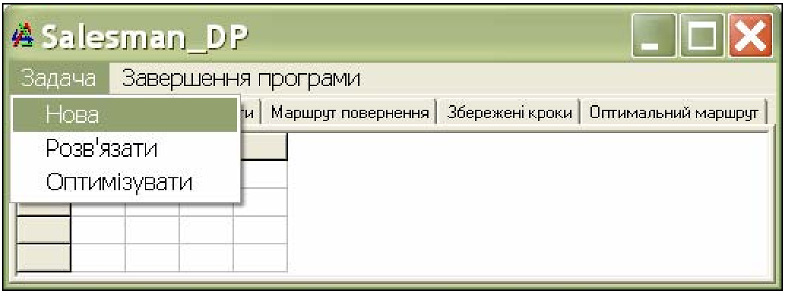


Рис. 4.2 – Вибір пункту меню для завантаження нової задачі

Після вибору пункту *Задача/Нова* на екрані дисплея з'являється діалогове вікно для введення вихідних даних задачі (рис. 4.3).

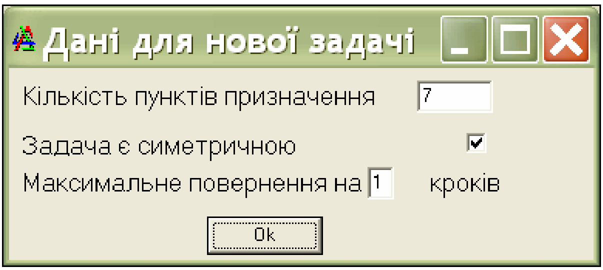


Рис. 4.3 – Діалогове вікно введення параметрів транспортної мережі і управління точністю рішення

У полі введення параметру *Кількість пунктів призначення* задається кількість пунктів транспортної мережі. У випадку, якщо відстані між будь-якими двома пунктами в прямому та зворотному напрямках руху збігаються, необхідно поставити галочку біля поля *Задача є симетричною*. Поле *Максимальне повернення* дозволяє регулювати точність отриманих рішень.

У табл. 4.1 надається залежність точності отриманих рішень від кількості кроків повернення. У ній також зазначено максимальну кількість пунктів транспортної мережі, для якої можна знайти рішення з відповідною точністю.

Максимальна кількість пунктів залежить не тільки від кількості кроків повернення, але й від параметрів ЕОМ, на якій працює програма. Тут і далі надаються результати тестування на ЕОМ із частотою процесора 1,50 ГГц і оперативною пам'яттю 256 Мб.

Таблиця 4.1 – Залежності середнього відносного відхилення від оптимального результату і припустимої кількості пунктів у транспортній мережі від кількості кроків повернення

Максимальна кількість кроків повернення	Середнє відхилення від оптимального результату	Максимальна кількість пунктів
0	7,00%	170
1	4,39%	100
2	3,09%	95
3	2,73%	90
4	1,20%	80
5	1,62%	75
6	1,81%	65

Не рекомендується задавати кількість кроків повернення більше за 4, тому що при цьому не тільки зменшується припустима кількість пунктів у транспортній мережі, але й втрачається ефект підвищення точності, тобто зменшення середнього відхилення від оптимального результату.

Після введення параметрів задачі та натискання кнопки *Ok* діалогове вікно закривається. Робиться обчищення таблиці відстаней, що знаходиться на закладці *Матриця відстаней*, від даних попередньої задачі та задаються розміри таблиць у відповідності до параметрів транспортної мережі.

Далі треба заповнити таблицю на закладці *Матриця відстаней*. Будемо вважати, що транспортна мережа містить n пунктів і всі пункти пронумеровані числами від 0 до $(n-1)$. Якщо усі відстані між пунктами симетричні, тобто довжина шляху між пунктами i та j не залежить від напрямку руху, то для кожної пари пунктів (i, j) досить увести потрібне число в клітинку, що знаходиться в i-му рядку і j-му стовпці таблиці ($i, j \in \{0,...,n-1\}$, $i \neq j$). У цьому випадку при виконанні наступних дій запускається процедура *SymmetryPerform*, за допомогою якої всі значення із заповнених клітинок таблиці копіюються в клітинки, що є симетричними до заповнених відносно головної діагоналі таблиці. У протилежному випадку довжина шляху з пункту i в пункт j вводиться в клітинку на перетинанні i-го рядка і j-го стовпця, а довжина шляху з пункту j в пункт i вводиться в клітинку на перетинанні j-го рядка і i-го стовпця. Тепер таке введення стає обов'язковим як для асиметричних, так і симетричних відстаней. У випадку, якщо між деякими пунктами k і l транспортний зв'язок відсутній, уводити що-небудь у відповідні клітинки не треба, $k, l \in \{0,...,n-1\}$. Всі клітинки, що залишилися порожніми, заповнюються автоматично при виконанні наступних процедур. Для такого заповнення в програмі передбачена процедура *CheckGrid*.

2.2) Пункт меню *Задача/Оптимізувати* (рис. 4.4) вибирається у випадку, якщо задані відстані не для всіх пар пунктів.

Рис. 4.4 – Вибір пункту меню для визначення маршрутів мінімальної довжини між усіма парами пунктів

За вибором пункту *Задача/Оптимізувати* обчислюються довжини маршрутів між тими пунктами, для яких безпосереднього транспортного зв'язку не існує. Побудова відповідних маршрутів виконується за допомогою убудованої процедури *OptimNet.*

Побудова маршрутів мінімальної довжини між пунктами, що безпосередньо між собою не зв'язані, здійснюється за допомогою матричного методу [5]. Докладний опис зазначеного методу наводиться в розділі 2 цієї монографії.

2.3) Пункт меню *Задача/Розв'язати* (рис. 4.5) вибирається для отримання шуканого маршруту комівояжера мінімальної довжини.

Рис. 4.5 – Вибір пункту меню для побудови маршруту комівояжера

Пункту *Задача/Розв'язати* відповідає процедура, що здійснює: підготовку даних і таблиць до розв'язання задачі; управління процесом розв'язання; відображення кінцевого результату на екрані дисплея. Ця процедура викликає допоміжну процедуру *Renew*. За допомогою процедури *Renew* виконується обчищення таблиць, що розташовані на закладках *Маршрути*, *Маршрут повернення* і *Збережені кроки*. Потім виконується перша ітерація – у верхню таблицю на закладці *Маршрути* вносяться маршрути. Кожний маршрут містить два пункти, перший із яких обов'язково має номер 0.

У нижню таблицю на цій же закладці вносяться довжини відповідних маршрутів. Уведені маршрути та їх довжини копіюються в таблицю на закладці *Збережені кроки*.

Далі $(n-1)$ разів виконується процедура *DynStep*. У цій процедурі реалізований один крок – одна ітерація методу динамічного програмування.

У процесі виконання процедури *DynStep* робиться виклик допоміжних процедур *DynStep*1 і *DynStepBack*. Також при її виконанні на кожному кроці здійснюється копіювання отриманих маршрутів у таблицю на закладці *Збережені кроки*.

За допомогою процедури *DynStep*1 на підставі результатів попередньої ітерації будуються маршрути $M_1, M_2, ..., M_N$, що закінчуються відповідно в пунктах 1, 2, …, *N*...

За допомогою процедури *DynStepBack* генеруються альтернативні маршрути.

Процедура *DynStepBack* вимагає введення двох параметрів *DepthBack* і *iNeedBack*. Дана процедура будує альтернативні маршрути для пункту з номером *iNeedBack*. На підставі маршрутів, отриманих на ітерації з номером на *(DepthBack*+1*)* менше поточної, будуються маршрути, що не містять пункт *iNeedBack*. Кількість пунктів у цих маршрутах поступово доводиться до значення, рівного кількості пунктів у маршрутах попередньої ітерації. На підставі отриманих альтернативних маршрутів будується маршрут $M'_{iNeedBack}$ найменшої довжини, що закінчується пунктом *iNeedBack*. Якщо за допомогою процедури *DynStep*1 маршрут $M_{iNeedBack}$ отримати не вдається, тобто довжина маршруту $M'_{iNeedBack}$ менше довжини маршруту $M_{iNeedBack}$, то маршрут $M'_{iNeedBack}$ зберігається як результат виконання поточної ітерації. У протилежному випадку зберігається маршрут $M_{iNeedBack}$.

Результатом циклічного виконання процедури *DynStep* є маршрути, що починаються з пункту 0 і проходять через усі пункти транспортної мережі рівно по одному разу. Після цього до кожного з них приєднується пункт 0, і визначаються довжини побудованих маршрутів. Серед отриманих кільцевих маршрутів вибирається той, що має найменшу довжину. Список пунктів цього маршруту заноситься в таблицю на закладці *Оптимальний маршрут*. Також на цю закладку заноситься довжина самого маршруту.

4.2 Опис програми, що реалізує алгоритм розбиття транспортної мережі на райони

Програма призначена для одержання розбивки транспортної мережі на райони й оцінки якості отриманої розбивки. Рекомендується застосовувати програму для транспортних мереж із кількістю пунктів понад 30 за умови, що всі споживачі обслуговуються одним постачальником. Усі пункти, що розподіляються по районах – адреси споживачів однорідного вантажу. Місцезнаходження постачальника (складу) не має значення.

Програма враховує:

- наявність природних перешкод між пунктами в межах міста (залізничних колій, річок і т. ін.), тобто суттєві різниці між відстанями по прямій і довжинами реальних маршрутів транспорту;
- нерівномірний розподіл споживачів на території міста, тобто різну щільність споживачів.

Назва програми – *Partition.exe*.

Програма написана в середовищі програмування *Delphi* 5 і працює під операційною системою *Windows*-95 (-98, -*XP*, -*NT*, -2000 і т.д.). Ніякого додаткового програмного забезпечення для запуску та виконання програми не потрібно. Редагування і рекомпіляція програми виконується також в середовищі програмування *Delphi* 5 і її більш пізніх версіях.

За допомогою даної програми вирішується така задача. Задану множину пунктів транспортної мережі необхідно розбити на декілька підмножин. Кожна підмножина являє собою множину пунктів, що належать одному із районів розбиття. Для кожної пари пунктів, що мають між собою безпосередній транспортний зв'язок, попередньо визначається відстань між ними.

Результатом виконання програми є підмножини пунктів, що визначають райони розбиття. Крім того, програма визначає розмір кожного району розбиття як довжину маршруту комівояжера.

Розбиття здійснюється таким чином, щоб розміри всіх районів незначно відрізнялися між собою.

У залежності від кількості пунктів у транспортній мережі, що необхідно розбити на райони, доводиться варіювати точність оцінки розмірів одержуваних районів. Зв'язок між кількістю пунктів і точністю, із якою можна знайти рішення задачі, описаний нижче.

Програма дозволяє отримувати розбивку транспортної мережі, що містить до 80 пунктів, на будь-яку кількість районів. При цьому, чим більше кількість районів, тим менше часу потрібно на виконання програми.

При розбитті мережі на порівняно невелику кількість районів (2 ÷ 5) за кінцевий час, варто відмовлятися від високої точності оцінки розмірів отриманих районів. Припустима кількість пунктів у транспортній мережі в залежності від заданої точності оцінки наведена в табл. 4.2.

Таблиця 4.2 – Зв'язок між обмеженнями на кількість районів у розбивці, кількістю пунктів транспортної мережі і точністю оцінки розмірів районів розбиття

Кількість районів розбиття	Максимальна припустима кількість пунктів в транспортній мережі при середньому відхиленні оцінки розмірів районів розбиття від оптимальної оцінки (d)				
	d=1,2%	d=2,7%	d=3,3%	d=3,6%	d=7%
2	40	100	130	150	200
3	60	130	160	180	240
4	140	180	210	240	300
7	200	230	260	280	320

У якості вихідних даних для програми розбиття задаються відстані між усіма парами пунктів. Якщо між деякими парами пунктів транспортний зв'язок не існує, то для цих пар треба побудувати маршрути мінімальної довжини. У програмі передбачене виконання такої операції.

Відстані між пунктами транспортної мережі повинні бути задані цілими числами в однакових одиницях виміру для усіх пар пунктів. Одиниці виміру треба вибирати таким способом, щоб середня відстань між парами пунктів не перевищувало числа 2147473647/(n + 1), де n – кількість пунктів транспортної мережі.

Алгоритм програми:

<u>1) Запуск основного вікна програми</u>

Основне вікно програми показано на рис. 4.6.

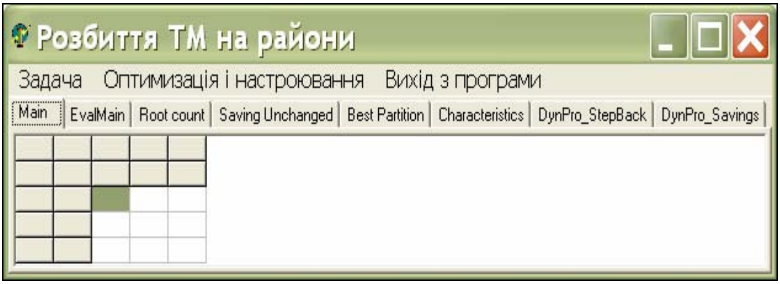


Рис. 4.6 – Основне вікно програми *Partition.exe*

На основному вікні програми є присутнім головне меню, а також декілька закладок. На кожній із закладок розташована таблиця.

Таблиця, що знаходиться на закладці *Main*, використовується для введення вихідних даних транспортної мережі і збереження зведеної матриці. У процесі формування районів в цій таблиці також зберігається проміжна інформація про склад кожного із районів.

У таблиці на закладці *EvalMain* зберігається матриця відстаней між пунктами. У ній, як і в таблиці на закладці *Main*, відбивається поточний склад районів на кожному етапі процесу розбиття. Таблиця використовується при оцінці розмірів отриманих районів.

У таблиці на закладці *Saving Unchanged* зберігаються вихідні дані. У таблицях на закладках *Main* і *EvalMain* початковий порядок пунктів змінюється, а на закладці *Saving Unchanged* залишається незмінним.

Таблиці на закладеннях *RootCount, DynPro_StepBack* і *DynPro_Savings* використовуються при оцінці розмірів районів. У таблицях зберігаються дані, що використовуються методом динамічного програмування для побудови маршруту комівояжера для кожного з районів.

У таблицях на закладках *Best Partition* і *Characteristics* після завершення процесу розбиття транспортної мережі на райони фіксуються результати, а саме: списки пунктів, що відносяться до кожного із районів, довжини маршрутів комівояжера для кожного із районів і їх сумарна довжина.

2) Виконання процедур, що запускаються з головного меню основного вікна програми

Для програмного розв'язання задачі розбиття рекомендується наступна послідовність дій:

2.1) Спочатку здійснюють уведення даних. У поля діалогового вікна (рис. 4.7), що відчиняється командою головного меню *Задача/Нова* (рис. 4.8), уводиться загальна кількість пунктів у транспортній мережі, необхідна кількість районів розбиття та вказується властивість несиметричності або симетричності усіх відстаней між пунктами

У програмі за допомогою процедури *Starting* робиться настроювання розмірів усіх таблиць відповідно до параметрів нової транспортної мережі. Крім того, процедура *Starting* передбачає обчищення таблиць від непотрібних даних.

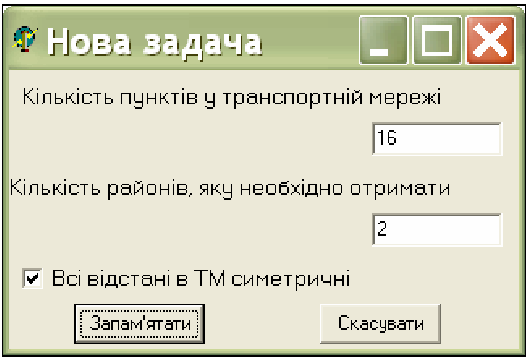


Рис. 4.7 – Діалогове вікно для введення параметрів транспортної мережі

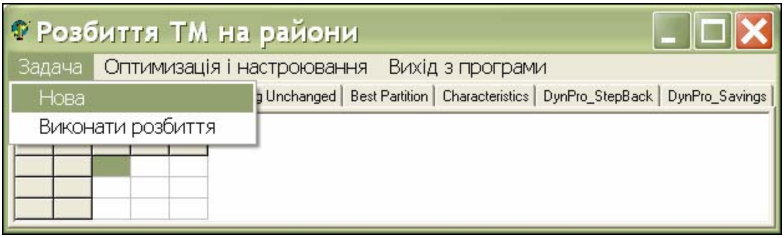


Рис. 4.8 – Вибір пункту меню для введення даних про нову задачу

Потім в основному вікні програми в таблицю на закладці *Main* уводяться відстані між пунктами. Уведення даних здійснюється таким же способом, як і в раніше описаній програмі *SlsManDProg.exe*.

2.2) Після введення вихідних даних виконується команда *Оптимізація і настроювання/Оптимізувати*, що запускає процедуру *OptimNet*. Ця команда та відповідна їй процедура виконуються тільки у випадку, коли відстані в попередньому пункті алгоритму були задані не для всіх пар пунктів.

Процедура *OptimNet* будує оптимальний маршрут для тих пар пунктів, відстань між якими не була зазначена, та обчислює довжину побудованого маршруту.

Процедура *OptimNet* ідентична процедурі з тією же назвою, що була описана раніше для програми *SlsManDProg.exe.*

Проміжні дані при виконанні процедури *OptimNet* зберігаються безпосередньо в таблиці на закладці *Main*. Після завершення процедури усі відстані між пунктами та довжини мінімальних маршрутів заносяться також у таблицю на закладці *EvalMain*.

2.3) Безпосереднє розбиття транспортної мережі на райони активується командою головного меню *Задача/Виконати розбиття* (рис. 4.9).

Рис. 4.9 – Вибір пункту меню для запуску процедури розбиття транспортної мережі на райони

Команда головного меню *Задача/Виконати розбиття* викликає процедуру *RegionForm*. Ця процедура видає запит на максимальну кількість кроків повернення в алгоритмі динамічного програмування та на припустиму різницю між районами у відносних одиницях (рис. 4.10).

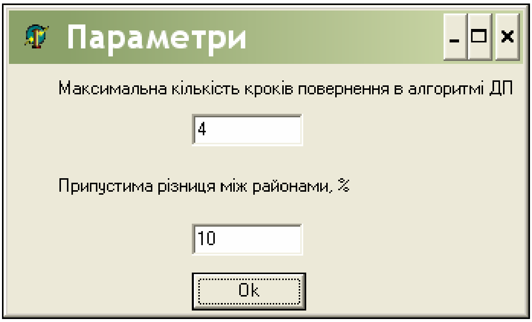


Рис. 4.10 – Форма введення параметрів для виконання розбиття

Як уже було сказано, максимальна кількість кроків повернення в алгоритмі динамічного програмування регулює точність оцінки отриманої розбивки. Найбільш точним результатам відповідає максимальне повернення на 4 кроки, найменш точним – на 0 кроків. Число кроків треба вибирати з урахуванням кількості пунктів у транспортній мережі. Обмеження на точність при заданій кількості пунктів наведені в табл. 4.2.

У силу дискретності розмірів отримати райони, різниця в розмірах яких не перевищує задану точність, іноді виявляється неможливим. Тому задана точність впливає на різницю розмірів районів, але не гарантує отримання зазначеної різниці в розмірах.

У таблиці 4.3 наведена залежність між заданими обмеженнями та отриманою в результаті тестування різниці між розмірами районів.

Таблиця 4.3 – Вплив заданих обмежень на різницю в розмірах, що отримана в результаті тестування районів

Тип фактично отриманої різниці	Задане обмеження, %				
	0	5	10	25	50
Середня різниця	23	23,7	24,6	28,9	39,5
Максимальна різниця	93,4	93,4	93,4	93,4	93,4
Мінімальна різниця	0,6	0,6	0,6	0,6	0,6

Відповідно до алгоритму, що був поданий в розділі 2 цієї монографії, здійснюється розбиття транспортної мережі на райони K способами, де K – задане число. Серед отриманих розбивок вибирається та, що має найменшу оцінку. По умовчанню число K задається рівним половині загальної кількості пунктів у транспортній мережі, тобто $K = \frac{N}{2}$. Однак це число може бути змінене вручну за допомогою пункту меню *Оптимізація і настроювання/Задати кількість* (рис. 4.11).

Рис. 4.11 – Вибір пункту меню для зміни кількості способів виконання розбиття

Після вибору пункту меню *Оптимізація і настроювання/Задати кількість* з'являється діалогове вікно для завдання величини K – кількості варіантів розбиття (рис. 4.12). Вікно дозволяє задати як число варіантів, що залежить від загального числа пунктів у транспортній мережі, так і будь-яке фіксоване число варіантів.

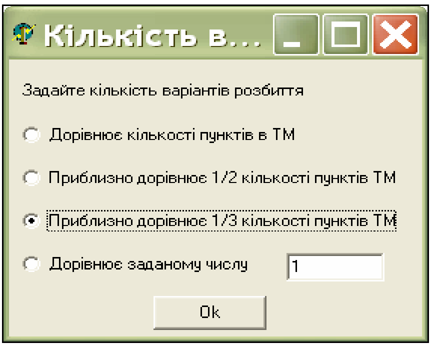


Рис. 4.12 – Завдання кількості варіантів розбиття

Задане число зберігається при розбитті всіх наступних транспортних мереж. У випадку, коли кількість варіантів розбиття було задано рівним, наприклад, $\frac{1}{3}$ від кількості пунктів транспортної мережі, то при розбитті нової транспортної мережі воно обновиться відповідно до її вимірності. Якщо ж було задано фіксоване число (обране *Дорівнює заданому числу*), то воно залишиться незмінним. Значення по умовчанню відновлюються тільки при повторному запуску програми.

Процедура *RegionForm* здійснює циклічний виклик процедури *PerfPartition*. Перед кожним викликом, крім першого, здійснюється копіювання даних із таблиці з закладки *Saving Unchanged* у таблицю на закладці *Main*. Оцінки отриманих розбивок порівнюються з найменшою оцінкою. Розбивка, що краще попередніх, зберігається. У таблиці на закладці *Best Partition* запам'ятовуються списки пунктів, що відносяться до кожного з районів найкращої розбивки. У таблиці на закладці *Characteristics* зберігаються розміри отриманих районів і номер найкращої розбивки.

У розділі 2 цієї монографії був описаний алгоритм модифікації методу «гілок і границь» для розбиття транспортної мережі на райони. Відповідно до алгоритму, слід було отримати декілька розбивок і серед них вибрати найкращу. Процедура *PerfPartition* реалізує частину алгоритму щодо здійснення одного розбиття транспортної мережі на райони. Ця процедура *PerfPartition* містить ряд допоміжних функцій і процедур. Список допоміжних функцій, процедур та їх призначення наводиться нижче.

Процедура *EvalMidDist* формує вектор середніх довжин транспортних зв'язків кожного з пунктів $\mathbf{L}_{cp}$.

За допомогою процедури *Deriving* робиться зведення матриці відстаней $\mathbf{D}$. У таблиці на закладці *EvalMain* зберігаються значення елементів матриці $\mathbf{D}$. Вони використовуються при обчисленні довжин маршрутів комівояжера. Елементи зведеної матриці $\mathbf{D}''$ містяться в таблиці на закладці *Main*. Потім ця таблиці використовується при формуванні районів розбиття.

Функція *DetRemote* вибирає перший пункт для району. Якщо формується перший район, то після вибору першого пункту району змінюються місцями перший рядок і перший стовпець, а також рядок і стовпець із номерами, що відповідають обраному пункту. Надалі, вибрані вже іншими процедурами пункти приєднуються до району таким же способом. Другі рядок і стовпець змінюються місцями з вибраними рядком і стовпцем, потім – треті і так далі.

Функція *NextNearestOut* визначає пункт, що є «найближчим» до даного району. Визначення пункту робиться по одному з критеріїв: (2.15) – (2.16), (2.17) – (2.19). Вибір того або іншого критерію відбувається в залежності від заданих параметрів функції.

Функція *NearestGroup* здійснює вибір району, до якого найбільш доцільно приєднати поточний пункт.

Процедура *DynamicProgr* визначає довжину маршруту комівояжера для заданого району. Застосувавши дану процедуру циклічно для всіх районів по черзі, можна обчислити сумарну довжину маршрутів комівояжера для всіх районів. У процедурі використовуються допоміжні процедури *DynStep*, *DynStep1*, *DynStepBack1*. Більш докладно призначення цих процедур було вже наведено для програми *SlsManDProg.exe*.

<u>3) Виведення отриманих результатів на екран</u>

Після закінчення процесу розбиття транспортної мережі на райони в таблицю на закладці *BestPartition* уміщуються списки пунктів, що відносяться до кожного з районів. У таблиці на закладці *Characteristics* знаходяться довжини маршрутів комівояжера для кожного з отриманих районів.

4.3 Приклад застосування програми розбиття транспортної мережі на райони

Розглянемо транспортну мережу, що містить 61 пункт. Пункти являють собою точки, що довільним способом розташовані на карті одного із районів м. Харкова. Карта району міста та розглянуті в прикладі пункти наведені на рис. 4.13.

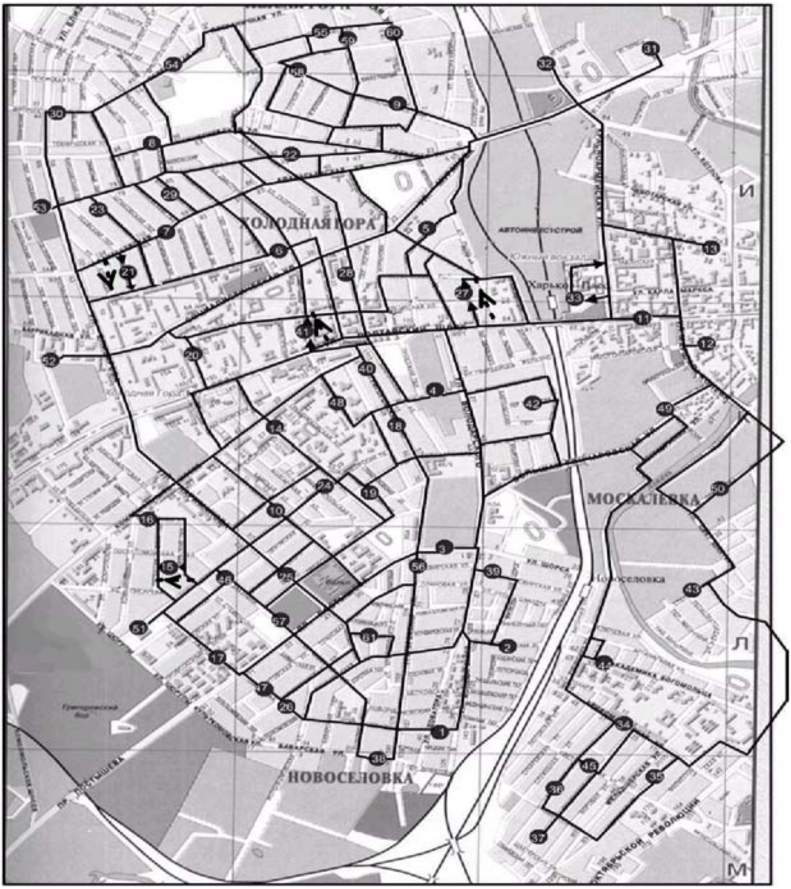


Рис. 4.13 – Карта одного із районів м. Харкова

Відповідно до вимог до вихідних даних програми, необхідно виміряти відстані між пунктами транспортної мережі. Відстані між пунктами вимірюються як довжини найкоротших маршрутів вдовж доріг міста.

Для скорочення кількості вимірів будемо вважати, що безпосередній транспортний зв'язок існує тільки між тими пунктами, на маршруті між якими не лежить ніякий інший пункт.

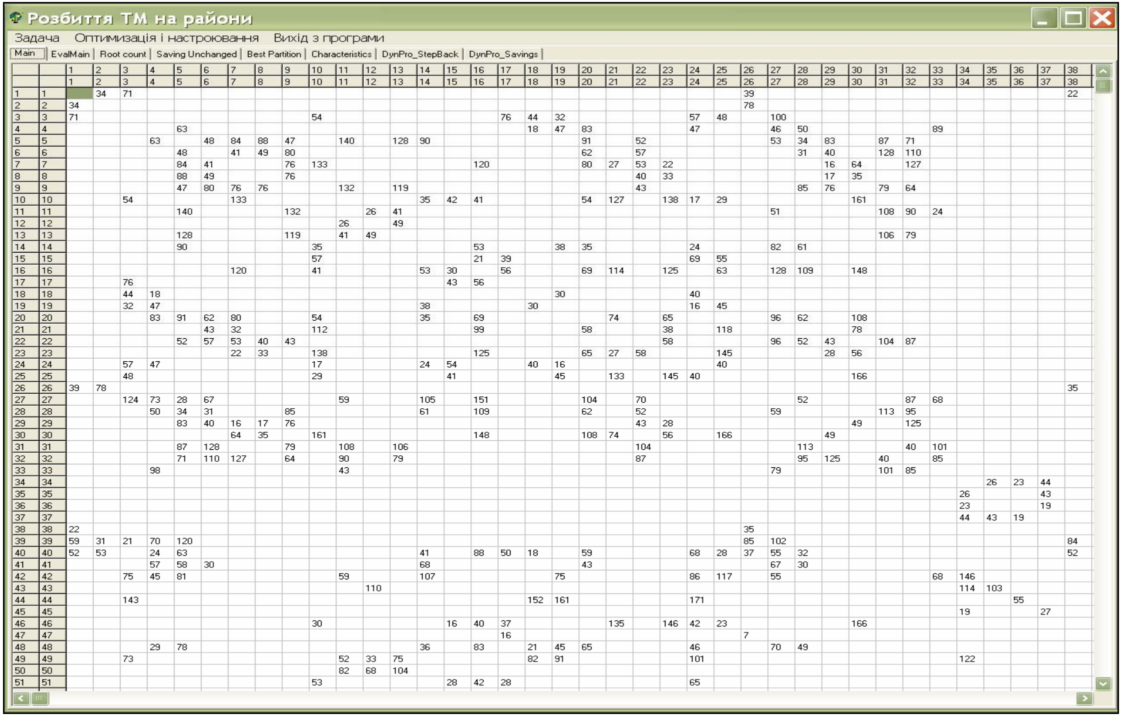
Для загальності прикладу будемо вважати, що транспортна мережа не є симетричною, тобто на деяких ділянках доріг рух є однобічним. Такі ділянки виберемо довільним способом.

На рис. 4.13 зазначені ділянки доріг, вдовж яких вимірювалися відстані. На тих ділянках, на яких рух був умовно заданий як однобічний, напрямок руху зазначений стрілками.

На рис. 4.14 зображено основне вікно програми *Partition.exe* із уже введеними відстанями між пунктами. Усі введені відстані відповідають довжинам ділянок доріг між пунктами на карті в міліметрах. При розв’язанні задачі в умовах прикладу суттєвими є тільки співвідношення відстаней між пунктами, тому масштаб карти не враховувався.

На рис. 4.15 зображено основне вікно програми вже після виконання команди *Оптимізація і настроювання/Оптимізувати*. Транспортна мережа подана в готовому до виконання розбиття вигляді.

В умовах прикладу було виконано розбиття транспортної мережі на 2 та на 3 райони. Кількість розглянутих варіантів розбивки склало приблизно половину загальної кількості пунктів транспортної мережі, тобто для обох випадків було розглянуто 31 варіант розбиття. Розбиття виконувалося при заданій припустимій різниці районів не більше 25%. При цьому середня точність оцінки отриманих районів становить 1,2%.



		1	2	3	4	5	6	7	8	9	10	11	12	13
		1	2	3	4	5	6	7	8	9	10	11	12	13
1	1		34	71										
2	2	34												
3	3	71									54			
4	4					63								
5	5				63		48	84	88	47		140		128
6	6					48		41	49	80				
7	7					84	41			76	133			
8	8					88	49			76				
9	9					47	80	76	76			132		119
10	10			54				133						
11	11					140				132			26	41
12	12											26		49
13	13					128				119		41	49	
14	14					90					35			
15	15										57			
16	16							120			41			
17	17			76										
18	18			44	18									
19	19			32	47									
20	20				83	91	62	80			54			
21	21						43	32			112			
22	22					52	57	53	40	43				
23	23							22	33		138			
24	24			57	47						17			
25	25			48							29			
26	26	39	78											
27	27			124	73	28	67					59		
28	28				50	34	31			85				
29	29					83	40	16	17	76				
30	30							64	35		161			
31	31					87	128			79		108		106
32	32					71	110	127		64		90		79
33	33				98							43		
34	34													
35	35													
36	36													
37	37													
38	38	22												
39	39	59	31	21	70	120								
40	40	52	53		24	63								
41	41				57	58	30							
42	42			75	45	81						59		
43	43												110	
44	44			143										
45	45													
46	46										30			
47	47													
48	48				29	78								
49	49			73								52	33	75
50	50											82	68	104
51	51										53			

		14	15	16	17	18	19	20	21	22	23	24	25	26
		14	15	16	17	18	19	20	21	22	23	24	25	26
1	1													39
2	2													78
3	3				76	44	32					57	48	
4	4					18	47	83				47		
5	5	90						91		52				
6	6							62		57				
7	7			120				80	27	53	22			
8	8									40	33			
9	9									43				
10	10	35	42	41				54	127		138	17	29	
11	11													
12	12													
13	13													
14	14			53			38	35				24		
15	15			21	39							69	55	
16	16	53	30		56			69	114		125		63	
17	17		43	56										
18	18						30					40		
19	19	38				30						16	45	
20	20	35		69					74		65			
21	21			99				58			38		118	
22	22										58			
23	23			125				65	27	58			145	
24	24	24	54			40	16						40	
25	25		41				45		133		145	40		
26	26													
27	27	105		151				104		70				
28	28	61		109				62		52				
29	29									43	28			
30	30			148				108	74		56		166	
31	31									104				
32	32									87				
33	33													
34	34													
35	35													
36	36													
37	37													
38	38													35
39	39													85
40	40	41		88	50	18		59				68	28	37
41	41	68						43						
42	42	107					75					86	117	
43	43													
44	44					152	161					171		
45	45													
46	46		16	40	37				135		146	42	23	
47	47				16									7
48	48	36		83		21	45	65				46		
49	49					82	91					101		
50	50													
51	51		28	42	28							65		

		27	28	29	30	31	32	33	34	35	36	37	38
		27	28	29	30	31	32	33	34	35	36	37	38
1	1												22
2	2												
3	3	100											
4	4	46	50					89					
5	5	53	34	83		87	71						
6	6		31	40		128	110						
7	7			16	64		127						
8	8			17	35								
9	9		85	76		79	64						
10	10				161								
11	11	51				108	90	24					
12	12												
13	13					106	79						
14	14	82	61										
15	15												
16	16	128	109		148								
17	17												
18	18												
19	19												
20	20	96	62		108								
21	21				78								
22	22	96	52	43		104	87						
23	23			28	56								
24	24												
25	25				166								
26	26												35
27	27		52				87	68					
28	28	59				113	95						
29	29				49		125						
30	30			49									
31	31		113				40	101					
32	32		95	125		40		85					
33	33	79				101	85						
34	34									26	23	44	
35	35								26			43	
36	36								23			19	
37	37								44	43	19		
38	38												
39	39	102											84
40	40	55	32										52
41	41	67	30										
42	42	55						68	146				
43	43								114	103			
44	44										55		
45	45								19			27	
46	46				166								
47	47												
48	48	70	49										
49	49								122				
50	50												
51	51												

Рис. 4.14 – Основне вікно програми *Partition.exe* з уведеними відстанями між пунктами

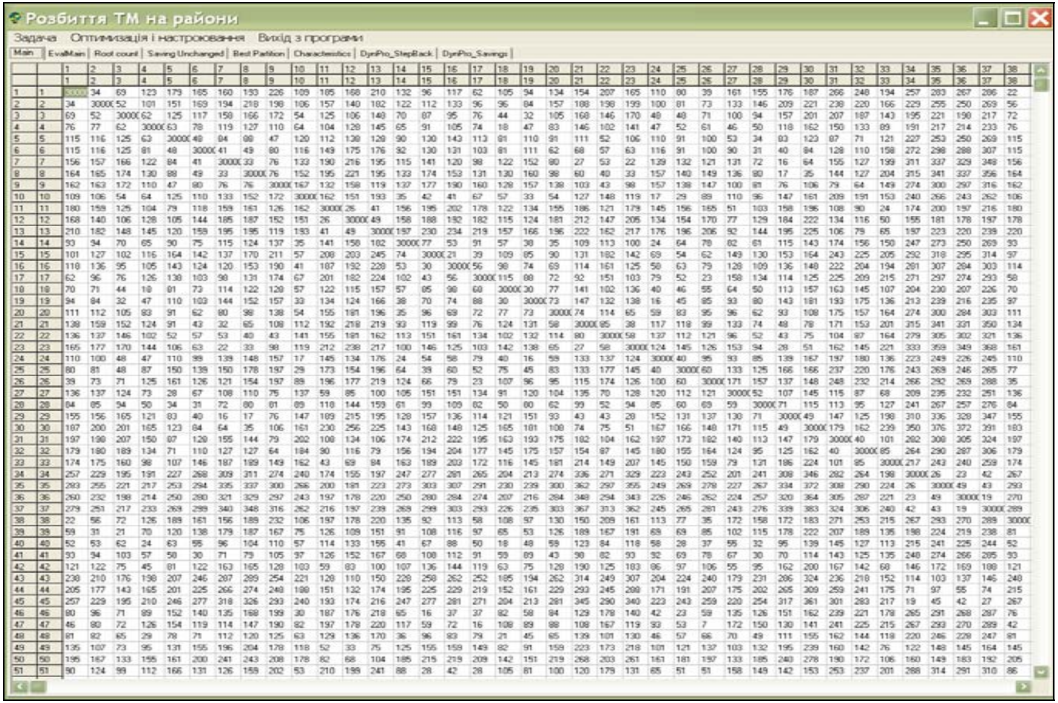
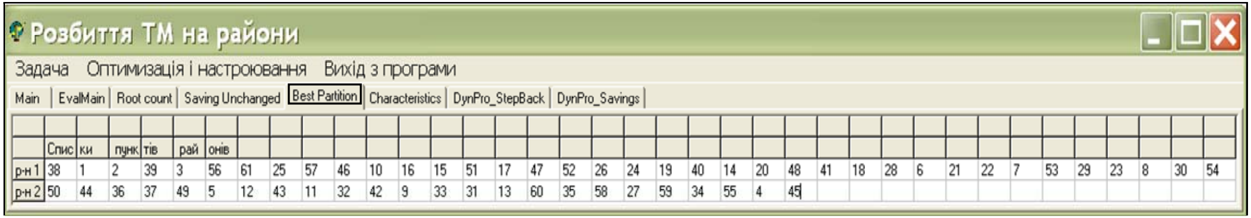
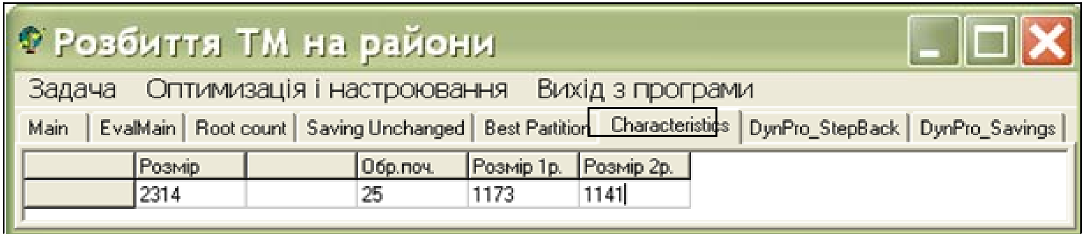


Рис. 4.15 – Основне вікно програми *Partition.exe* після виконання команди *Оптимізація і настроювання/Оптимізувати*



а) закладка *Best Partition*



б) закладка *Characteristics*

Рис. 4.16 – Результати розбивки транспортної мережі на 2 райони

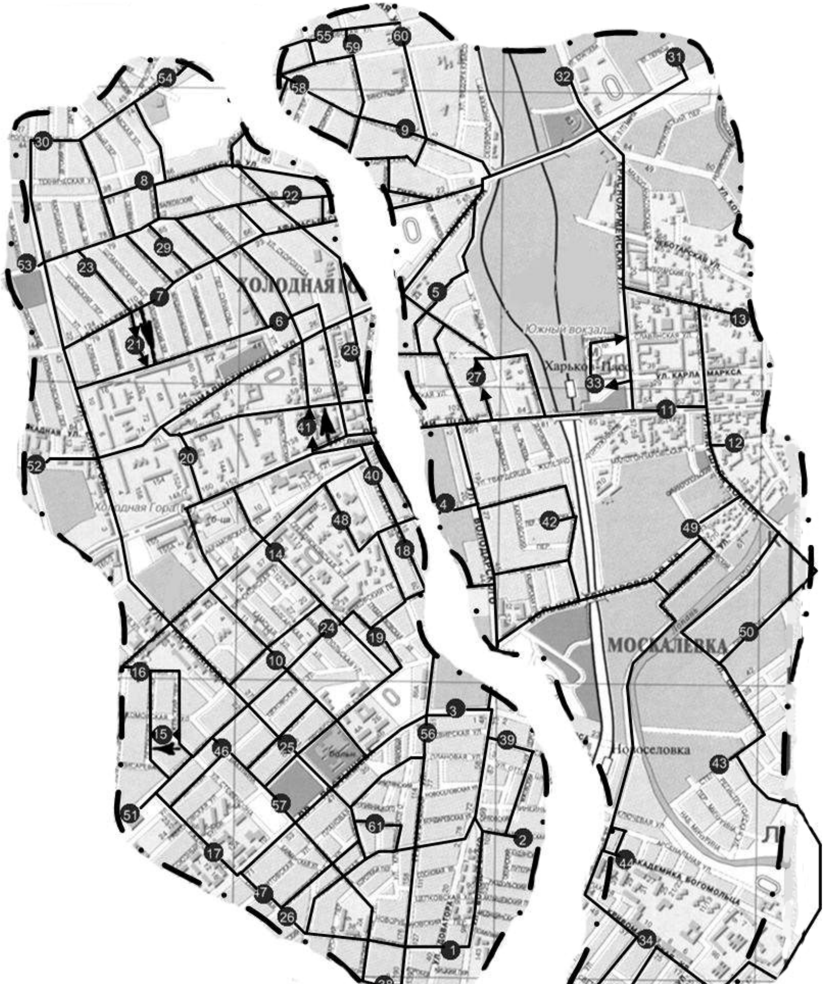


Рис. 4.17 – Результат розбивки транспортної мережі на 2 райони з прив'язкою до карти міста

На рис. 4.16 показані результати, що отримані програмою при розбитті транспортної мережі на 2 райони в умовах прикладу. Сформовані райони розрізняються своїми розмірами приблизно на 2,8%.

На рис. 4.17 результати розбиття показані на карті району міста. Очевидно, що площі всіх районів розбиття, відрізняються один від одного незначно, як і довжини відповідних маршрутів комівояжера.

На рис. 4.18 показані результати, що отримані програмою при розбитті транспортної мережі на 3 райони. Найбільша різниця між розмірами районів складає 5,1% від середнього розміру району.

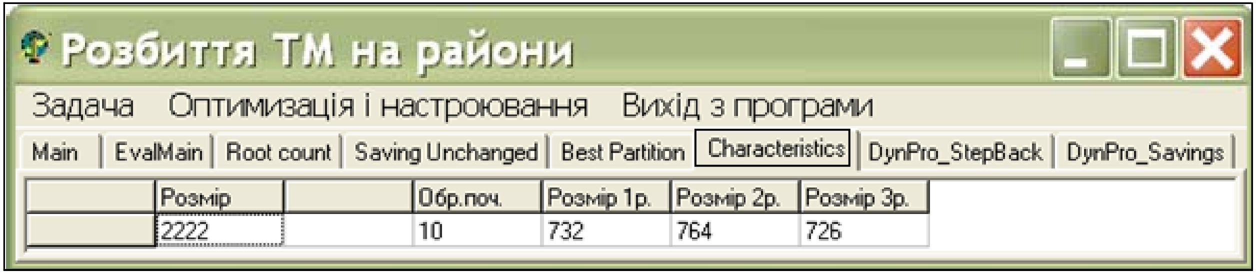
На рис. 4.19 показана карта району м. Харкова після розбиття відповідної транспортної мережі на три райони.

4.4. Висновки за результатами тестування програм, що реалізують модифікації методів «гілок і границь» і динамічного програмування

За допомогою тестування програм, що реалізують алгоритми модифікацій методів «гілок і границь» (програма *Salesman_DP.exe*) і динамічного програмування (програма *Partition.exe*), проведена оцінка їх практичної корисності.

Тестування програми *Salesman_DP.exe* показало, що при завданні різних вихідних даних для задачі комівояжера середня точність отриманих результатів становить від 1,2% до 7%. На порівняно застарілому персональному комп'ютері розв’язання такої задачі комівояжера можна здійснити для транспортних мереж, що містять до 170 пунктів.

а) закладка *Best Partition*



б) закладка *Characteristics*

Рис. 4.18 – Результати розбиття транспортної мережі на 3 райони

Рис. 4.19 – Результат розбиття транспортної мережі
на 3 райони з прив'язкою до карти міста

У результаті тестування програми *Partition.exe* на порівняно малопотужному комп'ютері було встановлено, що розбиття на два райони можна виконати для транспортної мережі, що містить до 80 пунктів, на три райони – до 90 пунктів, на чотири райони – до 100 пунктів. При наявності ж досить потужного комп'ютера розв’язання задачі розбиття транспортної мережі на потрібну кількість районів можна отримати для мереж досить великої вимірності.

Розроблені програми відповідно до цілей дослідження були орієнтовані на транспортні мережі великої вимірності, що мають місце у великих містах або регіонах із значним числом населених пунктів. Проведене тестування підтвердило доцільність їх використання в інженерній практиці саме для мереж великої вимірності.

Тестування також показало, що розроблений алгоритм адаптації методу «гілок і границь» для розбиття транспортних мереж на райони дозволяє робити розбиття з урахуванням особливостей місцевості та транспортних зв'язків між пунктами.

У процесі тестування було встановлено, що розроблена модифікація методу динамічного програмування не тільки дозволяє вирішувати задачу комівояжера великої вимірності, але й вирішувати її з високою точністю.

* * *

Автори будуть раді, якщо розглянуті моделі транспортних задач і методи розв’язання задач комівояжера великої вимірності або задач розбиття транспортної мережі великої вимірності на задане число районів будуть корисними читачам або послужать для них вдалою відправною точкою в розробці нових моделей і методів.

Автори будуть вдячні усім, хто запропонує нові, більш універсальні засоби розв’язання розглянутих задач або удосконалить старі. Розробка таких засобів – благодатний засіб для молодих дослідників, що бажають спробувати свої сили на ниві науки.

Теорія транспортних перевезень чекає своїх дослідників!

СПИСОК ДЖЕРЕЛ

1. Транспортная логистика [Текст] / Н.Э.Тамбаева, В.Л.Гудкова; под общ. ред. Л.Б. Миротина. – М.: «Экзамен», 2003. – 512 с.

2. Смехов, А.А. Введение в логистику [Текст] / А.А. Смехов. – М.: «Транспорт», 1993. – 112 с.

3. Хоменко, Л.М. Підвищення ефективності транспортних послуг в умовах трансформаційної економіки [Текст] / Л.М. Хоменко // Регіональні перспективи. – Кременчук, 2003, № 7 – 8. – С. 78 – 80

4. Лукинский, В.С. Логистика: [Электрон. ресурс]. – Режим доступа: http: // www.globalteka.ru/ articles /docdetails /6774---.html

5. Житков, В.А. Методы оперативного планирования грузовых автомобильных перевозок [Текст] / В.А. Житков, К.В. Ким. – М.: Транспорт, 1982. – 184 с.

6. Бочкарев, А.А. Решение задачи оптимизации доставки мелкопартионных грузов в условиях крупного города методом локализации [Текст] / А.А., Бочкарев, О.Н. Анисимова // Логистика сегодня. – 2008. – № 3. – С. 162 – 181.

7. Ломко, Е. Автоматизация логистических процессов предприятия как один из действенных инструментов преодоления кризиса [Электрон. ресурс]. – Режим доступа: http: // consulting.1c.ru/articles-view.jsp?id=44

8. Онлайн-конференция: оптимизация транспортной логистики [Электрон. ресурс]. – Режим доступа: http: // www.ukrbiznes.com/analitic/electronic/4621. html

9. Воркут, А.И. Грузовые автомобильные перевозки [Текст] / А.И. Воркут. – К.: «Вища школа», 1986. – 447 с.

10. Шептура, А.Н. Повышение эффективности автомобильных перевозок партионных грузов при переменном спросе на перевозки [Текст]: дисс. … канд. техн. наук / А.Н. Шептура. – Х., 2004. – 150 с.

11. A Tutorial on Clustering Algorithms. Fuzzy C-Means Clustering // http://home.dei.polimi.it/matteucc/Clustering/tutorial html/cmeans.html

12. Гончаров, М. Кластеризация на основе нечетких отношений [Электрон. ресурс]. – Режим доступа: http: // www.spellabs.ru/FuzzyRelationClastering.htm

13. Теория и методология процессного подхода к моделированию и интегрированному планированию цепи поставок[Текст]: автореф. дисс. … д-ра экон. наук: 08.00.05 / Бочкарев А.А.; Санкт-Петербург. экон. ун-т. – СПб., 2009. – 39 с.

14. Дискретное программирование [Текст] / А.А. Корбут, Ю.Ю. Филькенштейн; под общ. ред. Юдина Д.Б. – М.: Наука, 1969 – 368 с.

15. Беллман, Р. Применение динамического программирования к задаче о коммивояжере [Текст] / Р. Беллман // Кибернетический сборник. – М.: Мир, 1964. – Вып. 9. – С. 219 – 222.

16. Куликов, А. Алгоритмы и структуры данных. Лекция 7: Динамическое программирование [Электрон. ресурс]. – Режим доступа: http: // logic. pdmi.ras.ru/~kulikov/ simplealgscourse/lecture7-slides.pdf С. 96

17. Зарецкий, Л.С. Решение задачи коммивояжера и задач развозки методом коррекции функций состояния [Текст] / Л.С. Зарецкий // Методы оптимизации перевозочного процесса на автотранспорте. – М., 1976. С. 70 – 83.

18. Алгоритм решения задачи коммивояжера [Текст] / Дж. Литтл и др. // Экономика и математические методы. – 1965. - №1. – С. 94 – 107.

19. Крушевский, А.В. Задача о бродячем торговце [Текст]: / А.В. Крушевский // Материалы науч. семинаров по теоретическим и прикладным вопросам кибернетики. – К.: Освіта, 1964. – 320 с.

20. Foster B.A., Ryan D.M. An integer programming approach to the vehicle scheduling problem. // Operational Research Quarterly. – 1976. – V.4, №1. – P. 367 – 384.

21. Beltrami E., Bodin L. Networks and vehicle routing for municipal waste collection. // Networks. – 1974. – V.4, №1. – P. 65 – 94.

22. Evans S.R., Norback J.P. The impact of a decision-support system for vehicle routing in foodservice supply situation. // Journal of the Operational Society. – 1985. – V.36, №4. – P.467 – 472.

23. Чалый, А. Ситуационные методы планирования и управления перевозками мелкопартионных грузов [Текст] / А. Чалый, Б. Рыбак // Автомобильный транспорт. – 1983. - №9. – с. 16 – 19.

24. Christofides N., Eilon S. Algorithm for large-scale traveling salesman problem. // Operational Research Quarterly. – 1972. – V. 23, №4. – P.511 – 518.

25. Groes G. A method for solving traveling salesman problems. // Operations Research. – 1958. – V.6, №5. – P. 791 – 812.

26. Russel L. An effective heuristic for M-tour traveling salesman problem with some side condition. // Operations Research. – 1977, V. 25, №3. – P. 517 – 524.

27. Ефремов, А.В. Совершенствование планирования перевозок и их исследование [Текст] / А.В. Ефремов, Л.Б. Трахтенберг // Методы исследования операций в задачах автомобильного транспорта. – М.: МАДИ, 1983. – С. 4 – 9.

28. Gillett B., Miller L. A heuristic algorithm for the vehicle dispatch problem. // Operations Research. – 1974. – V. 22, №3. – P. 340 – 349.

29.Miller G.E., Tucker A.W., Zemlin R.A. Integer programming formulation of traveling salesman problems. // Computer Machinery. – 1960. V. 7, №4. – P. 326 – 329.

30.Clarck G., Wright J. Scheduling of vehicles from a central depot to a number of delivery points. // Operations Research. – 1964. – V. 12, №4. – P. 568 – 581.

31.Gaskell T.I. Bases for vehicle fleet scheduling. // Operations Research Quarterly. – 1967. V. 18, №2. – P. 281 – 295.

32.Golden B., Magnanti I., Nguyen H. Implementing vehicle routing algorithm. // Networks. – 1977. – V.7, №2 – P. 113 – 148.

33.Holmes R.A., Parker R.G. A vehicle scheduling procedure based upon savings and solution perturbation scheme. // Operations Research Quarterly. – 1976. – V. 27, №1. – P. 83 – 92.

Наукове видання

САМОЙЛЕНКО Микола Іванович,
КОБЕЦЬ Анна Олександрівна

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В РОЗВ'ЯЗАННІ ТРАНСПОРТНИХ ЗАДАЧ

МОНОГРАФІЯ

За авторською редакцією
Комп'ютерне верстання *М. І. Самойленко*
Дизайн обкладинки *І. П. Шелехов*
Технічне редагування *А.О. Кобець*

Підп. до друку 04.03.11
Формат 60x84 1/16
Тираж 300 пр.
Друк на ризографі
Ум. друк. арк. 16,0
Зам. №

Видавець і виготовлювач:
Харківська національна академія міського господарства,
вул. Революції, 12 Харків, 61002
Електронна адреса: rectorat@ksame.kharkov.ua
Свідоцтво суб`єкта видавничої справи: ДК №731
від 19.12.2001